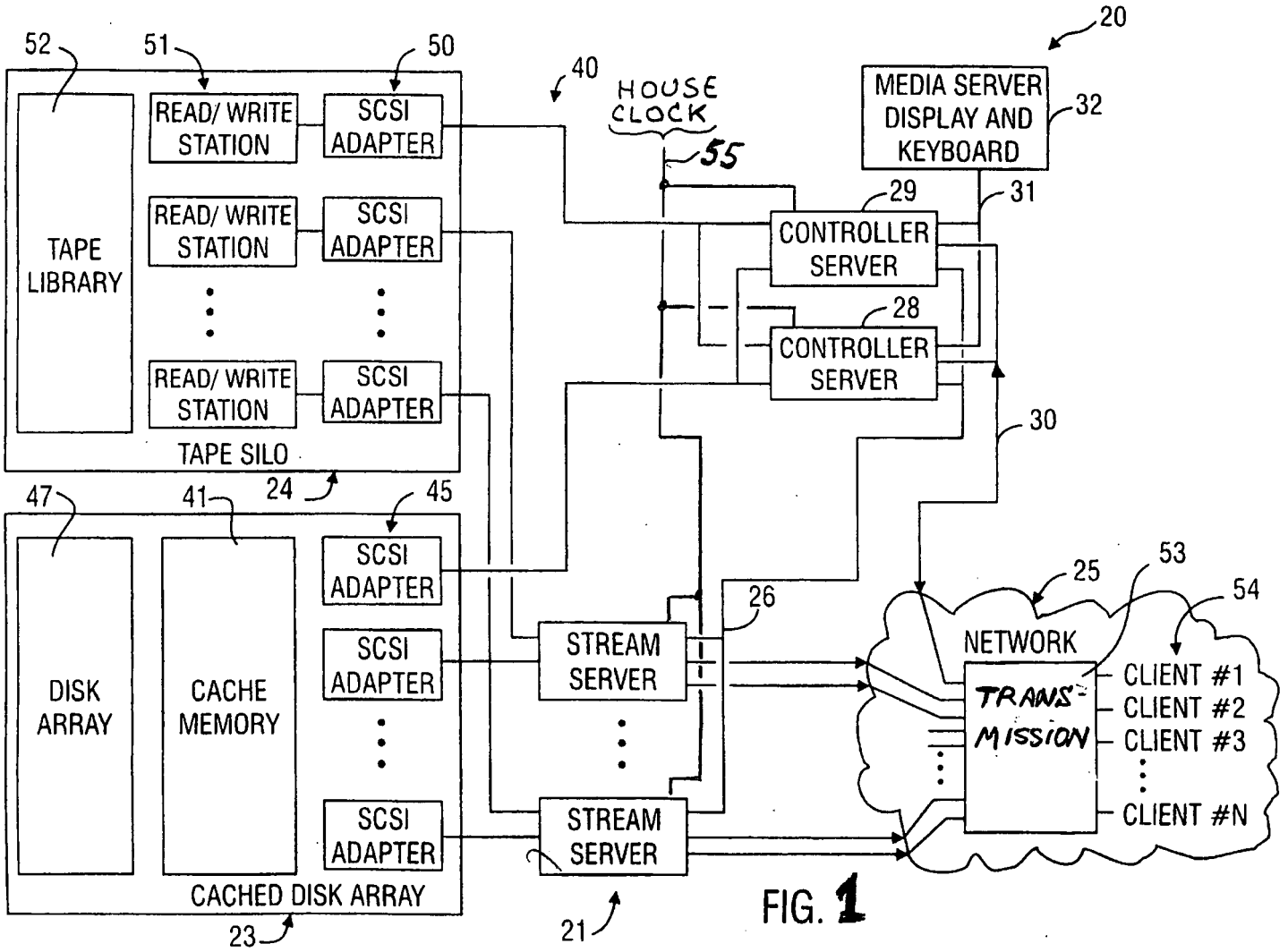

FIG. 1

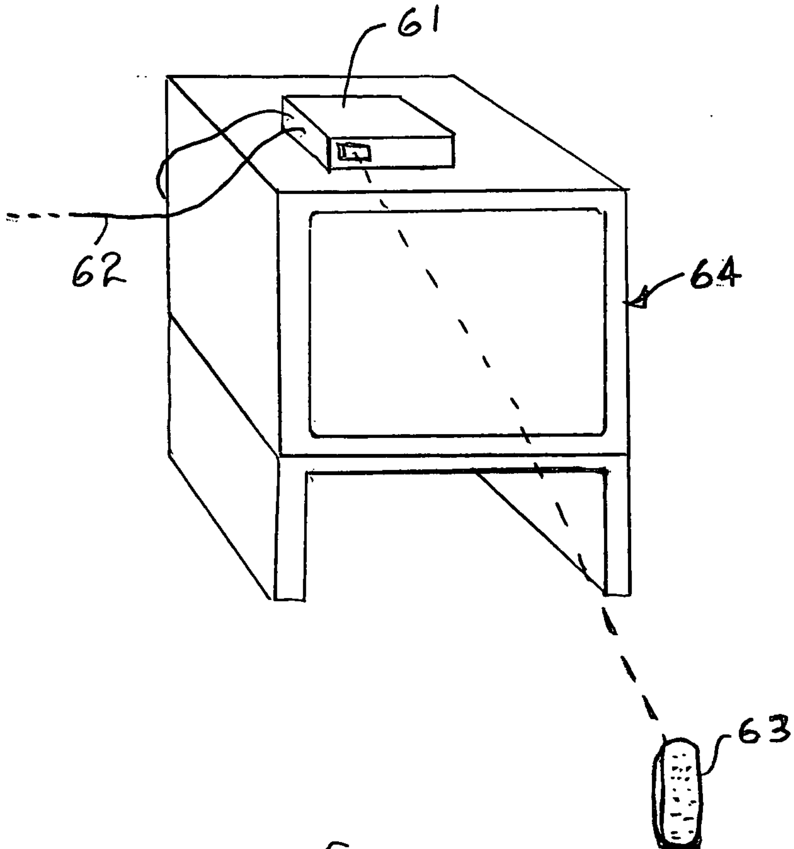
61
62
64
63

FIG. 2

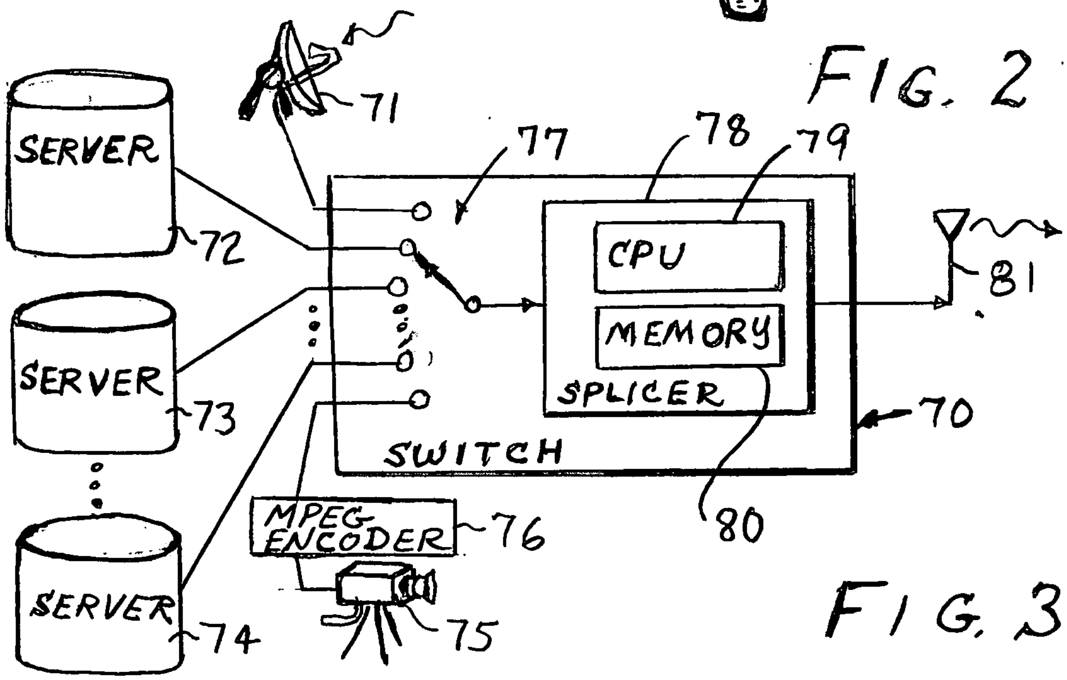
71
SERVER
72
SERVER
73
SERVER
74
77
78
79
CPU
MEMORY
SPLICER
SWITCH
81
70
80
MPEG ENCODER
76
75

FIG. 3

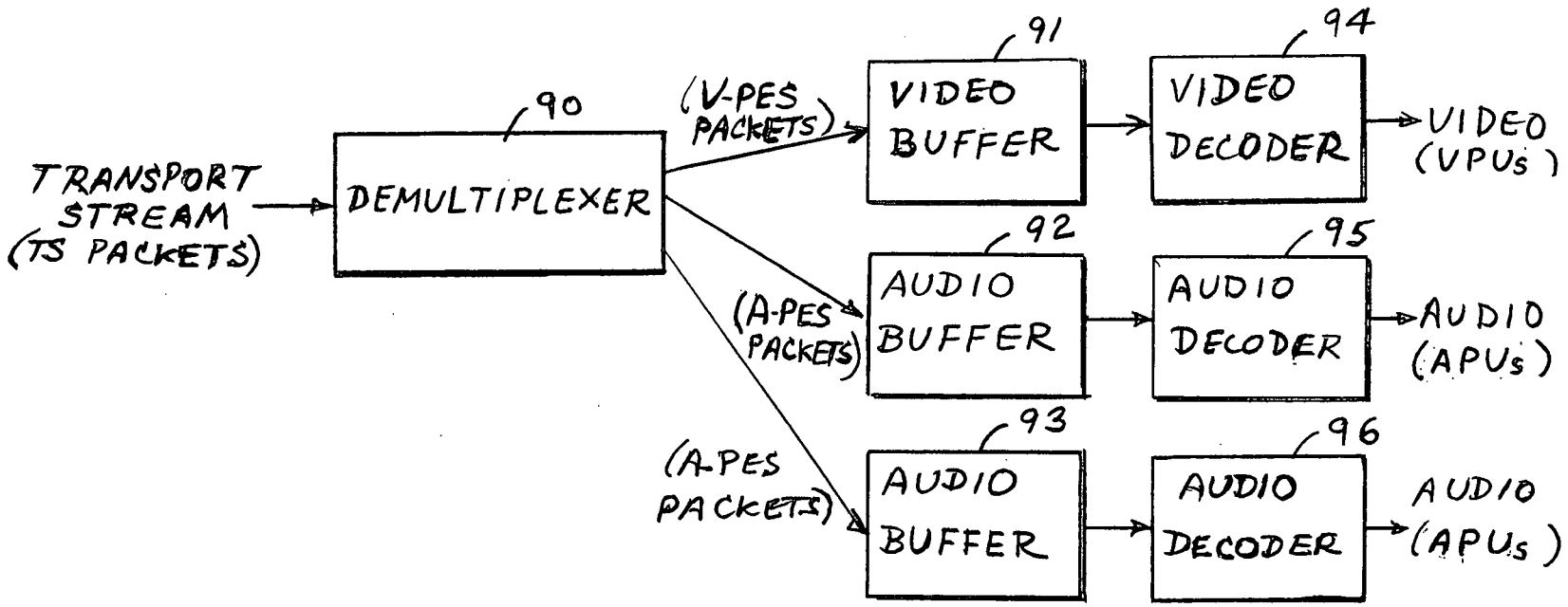

FIG. 4

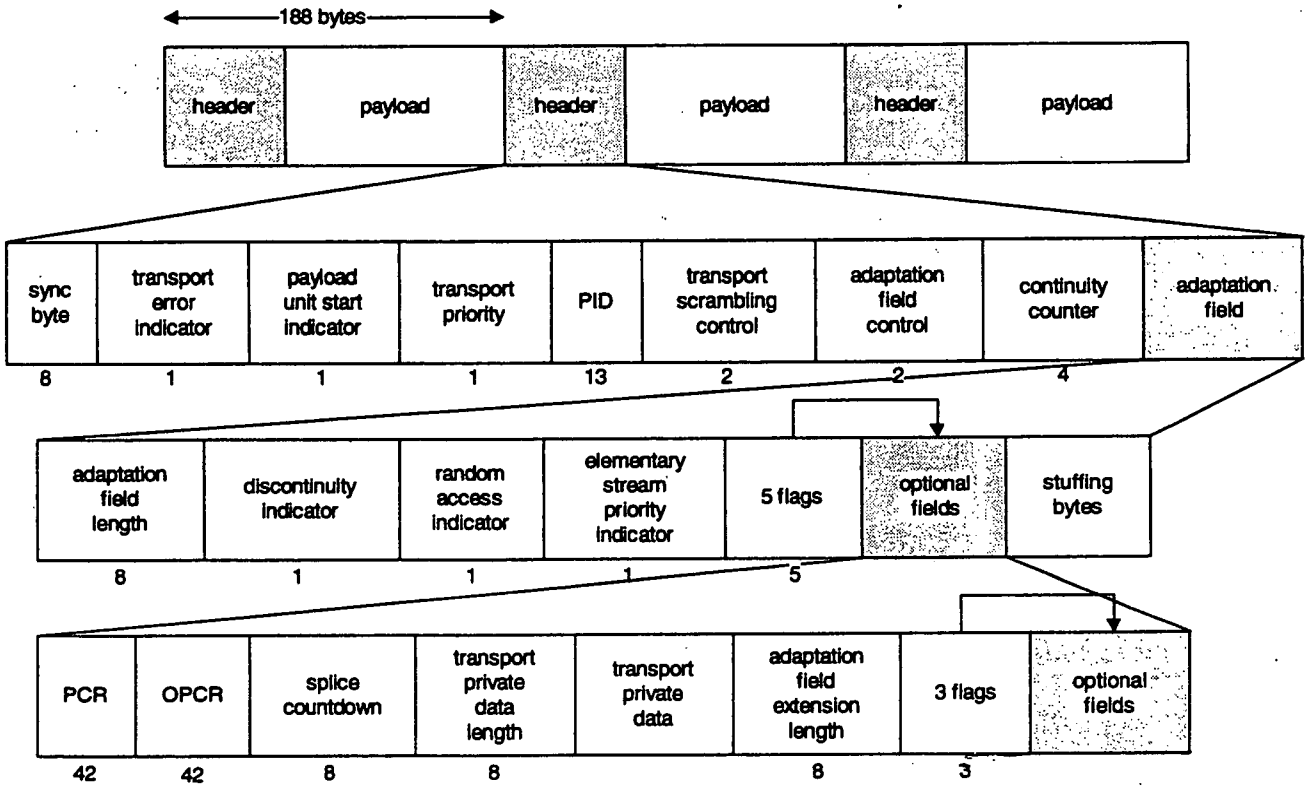

FIG. 5

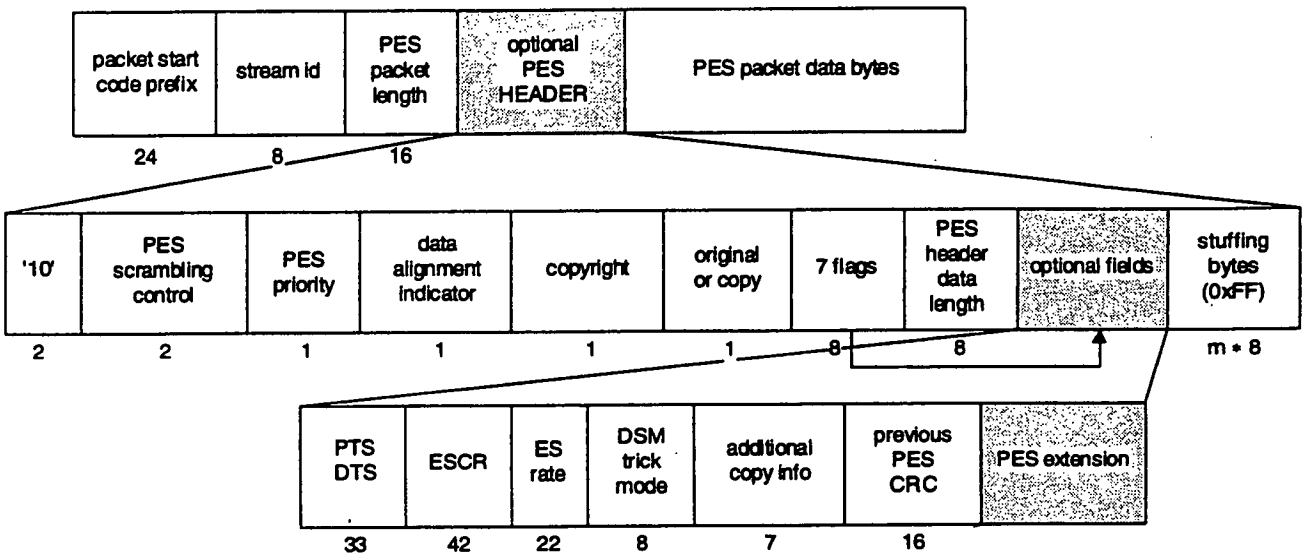

FIG. 6

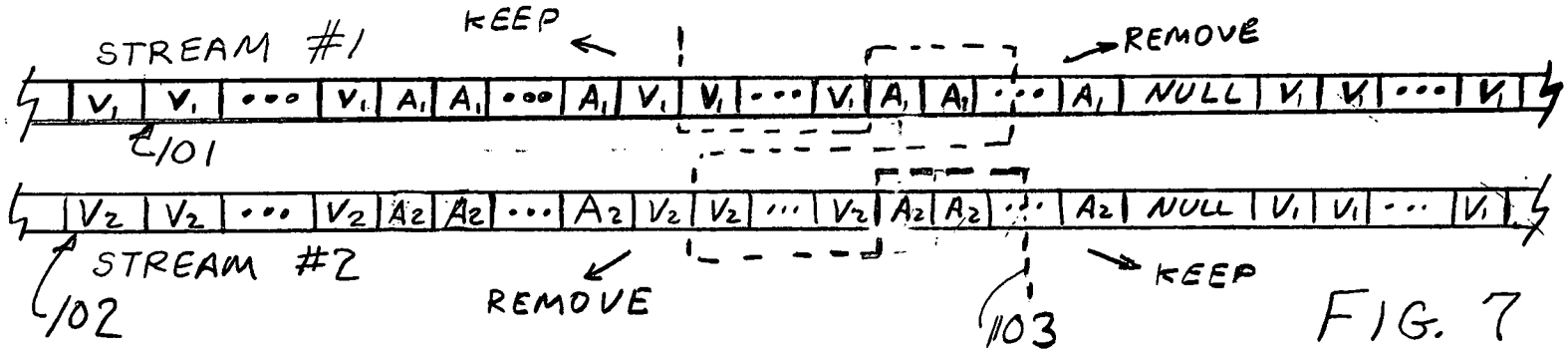

FIG. 7

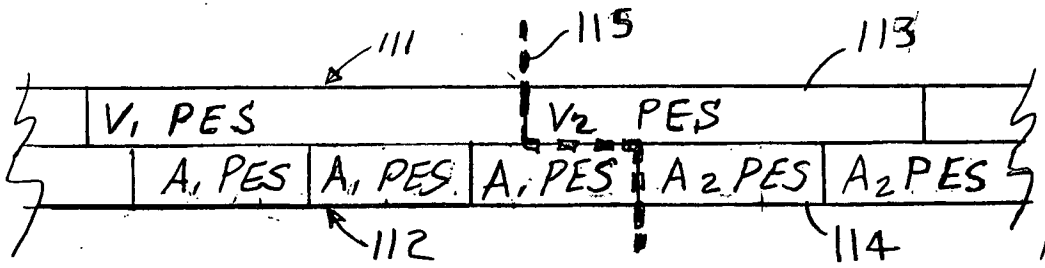

FIG. 8

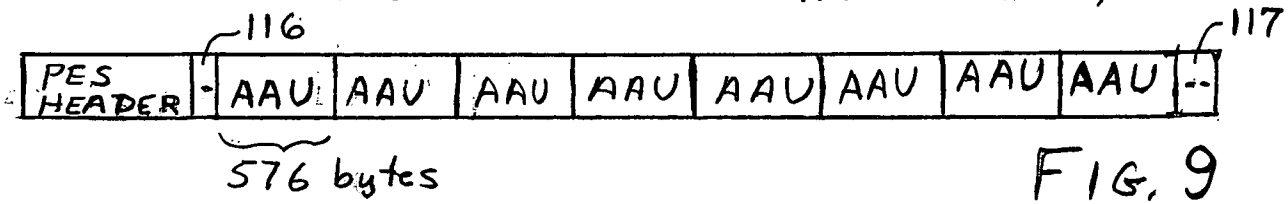

FIG. 9

STREAM #1 BEST ALIGNED APU SHORT INTO THE CUT ($\Delta_1 > 0$)	STREAM #2 BEST ALIGNED APU SHORT INTO THE CUT ($\Delta_2 < 0$)	12 msec. < audio gap < 24 msec. ($\Delta_1 - \Delta_2$)	FIGS. 11A, 11B, 11C
		0 msec. < audio gap < 12 msec. ($\Delta_1 - \Delta_2$)	FIGS. 12A, 12B
	STREAM #2 BEST ALIGNED APU LONG INTO THE CUT ($\Delta_2 > 0$)	0 msec. < audio gap < 12 msec. ($\Delta_1 - \Delta_2$)	FIGS. 13A, 13B
		0 msec. < audio overlap < 12 msec. ($\Delta_2 - \Delta_1$)	FIGS. 14A, 14B
STREAM #1 BEST ALIGNED APU LONG INTO THE CUT ($\Delta_1 < 0$)	STREAM #2 BEST ALIGNED APU SHORT INTO THE CUT ($\Delta_2 < 0$)	0 msec. < audio gap < 12 msec. ($\Delta_1 - \Delta_2$)	FIGS. 15A, 15B
		0 msec. < audio overlap < 12 msec. ($\Delta_2 - \Delta_1$)	FIGS. 16A, 16B
	STREAM #2 BEST ALIGNED APU LONG INTO THE CUT ($\Delta_2 > 0$)	12 msec. < audio overlap < 24 msec. ($\Delta_2 - \Delta_1$)	FIGS. 17A, 17B, 17C
		0 msec. < audio overlap < 12 msec. ($\Delta_2 - \Delta_1$)	FIGS. 18A, 18B

FIG. 10

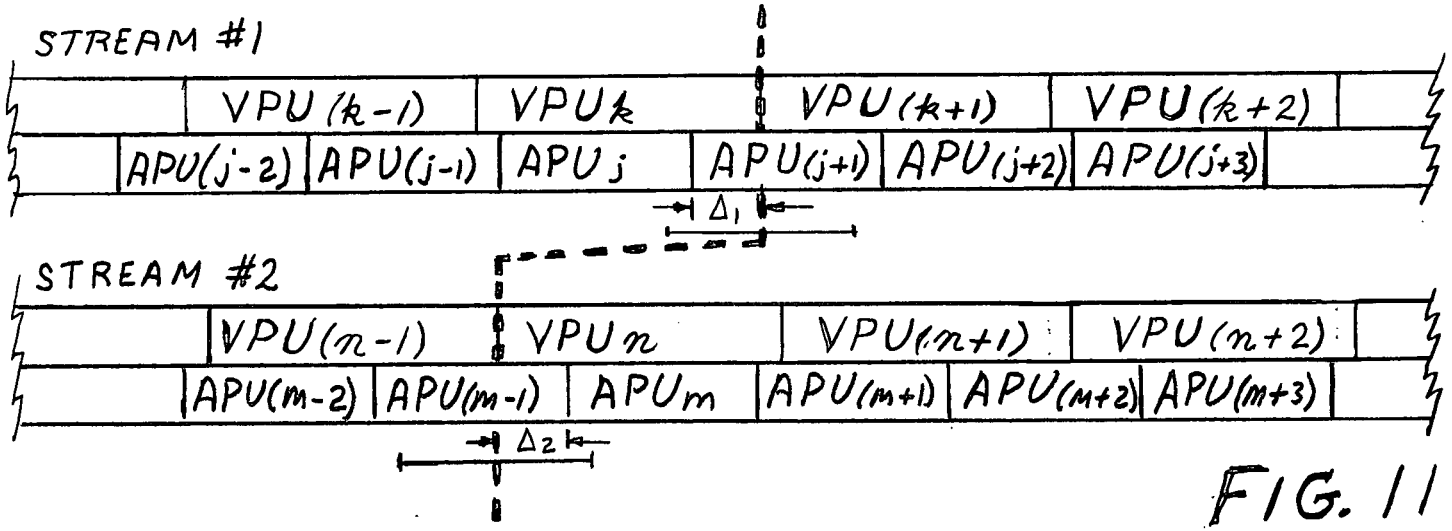

FIG. 11A

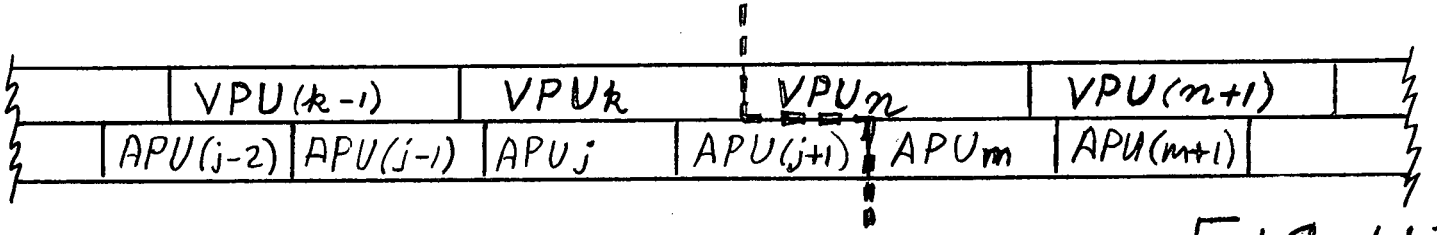

FIG. 11B

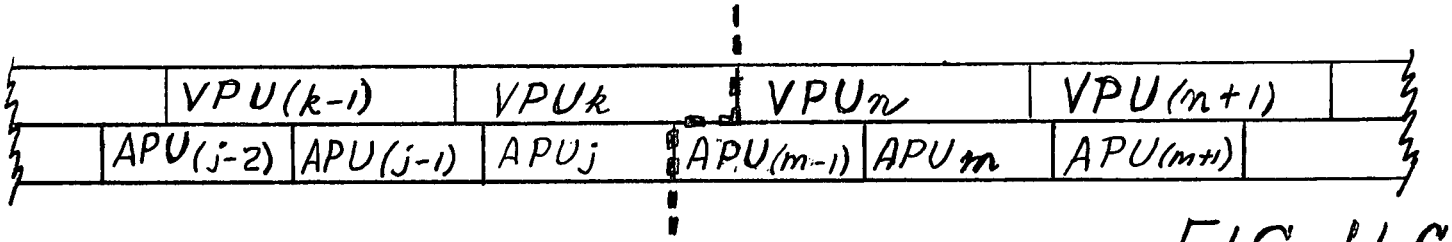

FIG. 11C

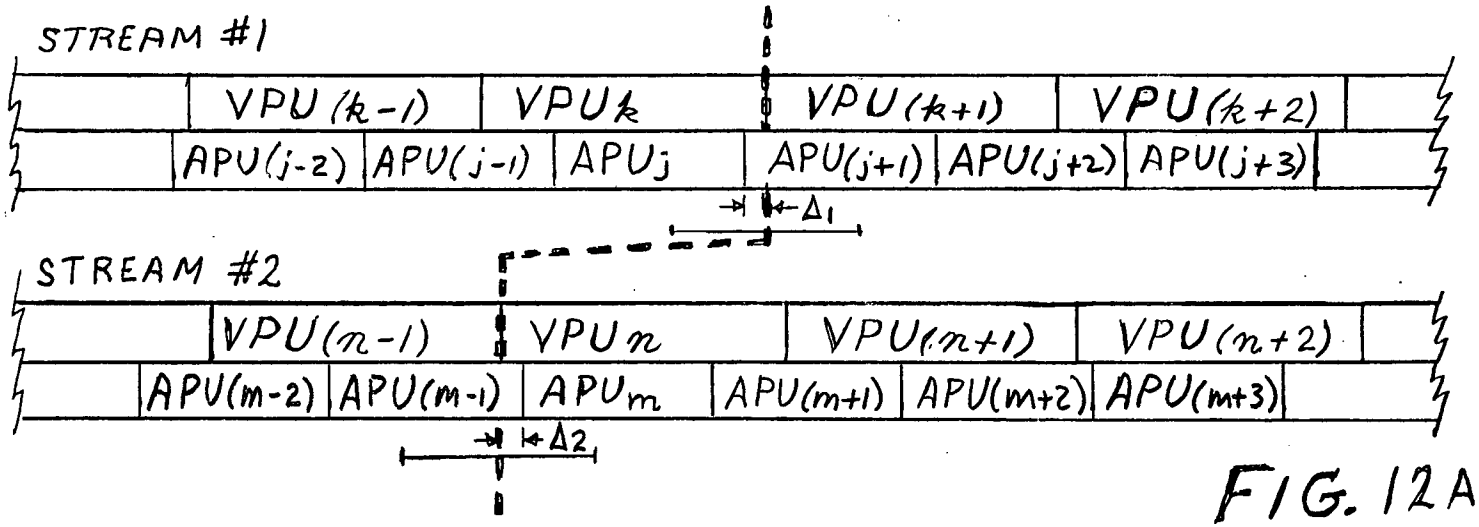

FIG. 12A

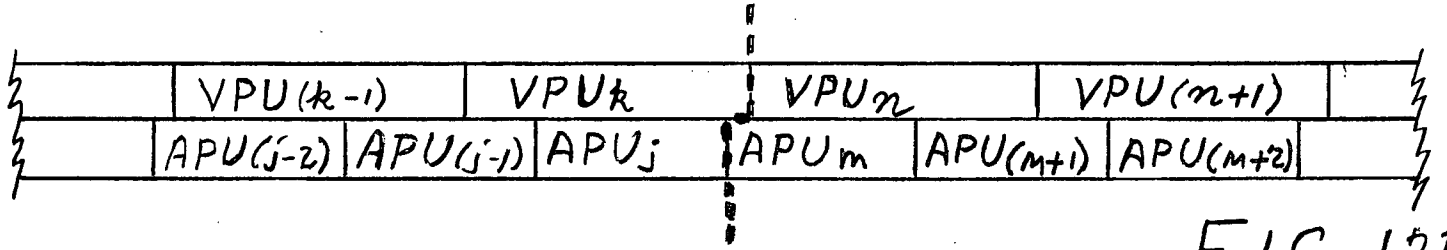

FIG. 12B

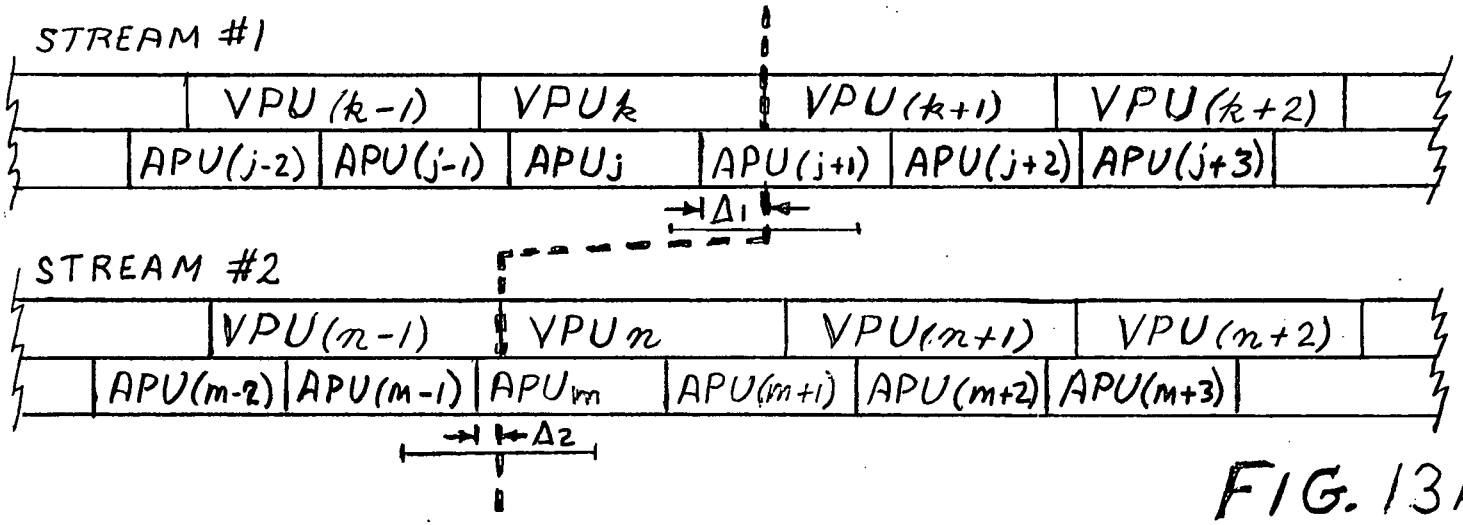

FIG. 13A

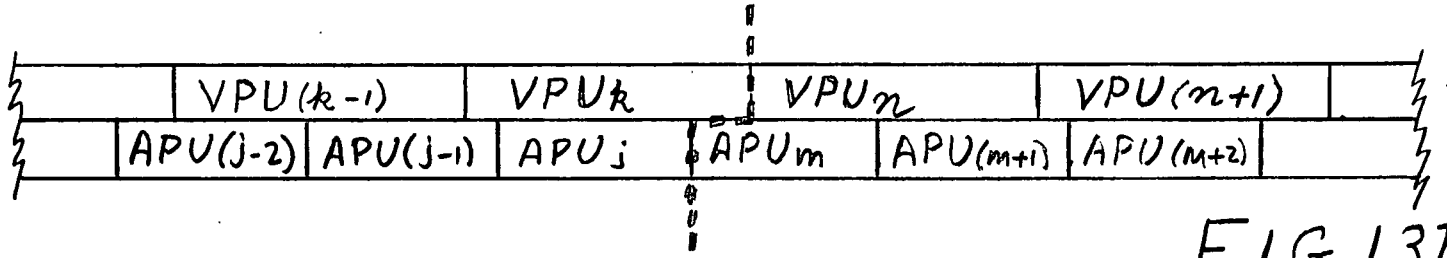

FIG. 13B

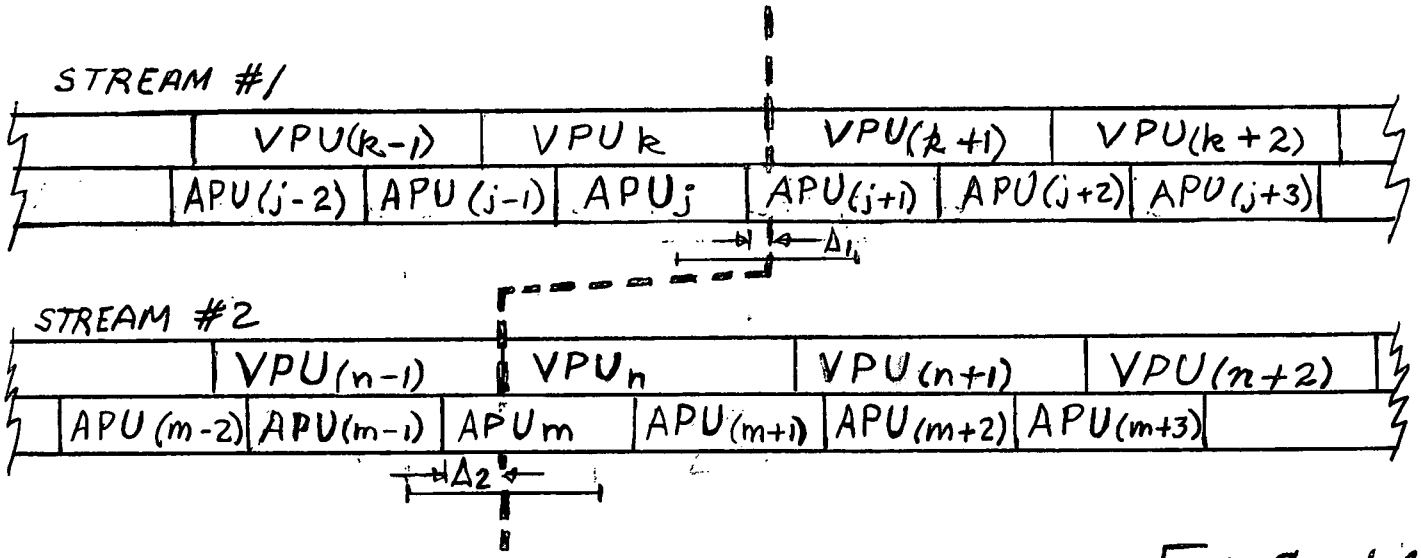

FIG. 14A

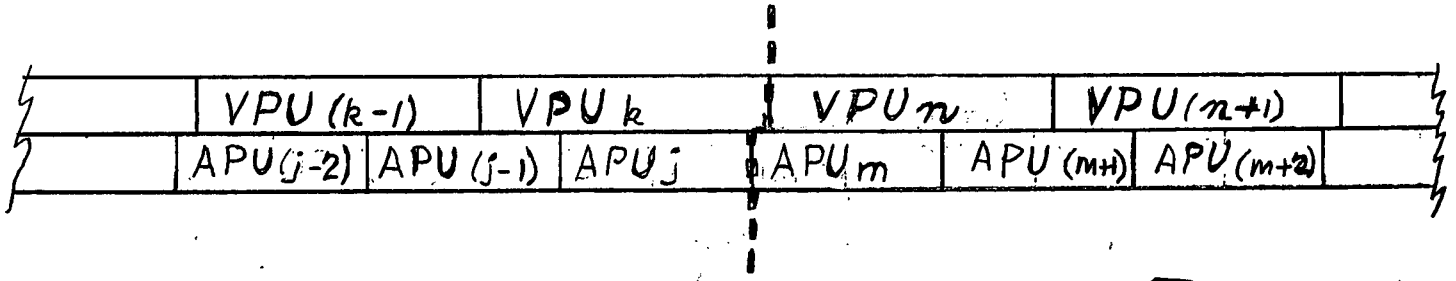

FIG. 14B

Case 5a

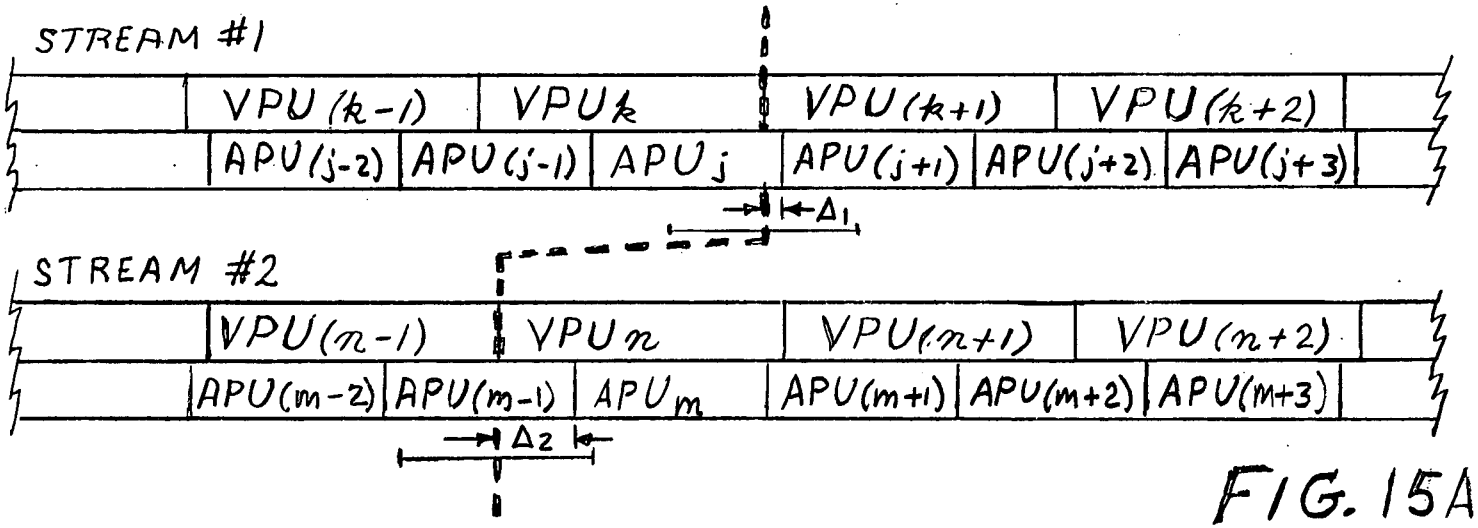

FIG. 15A

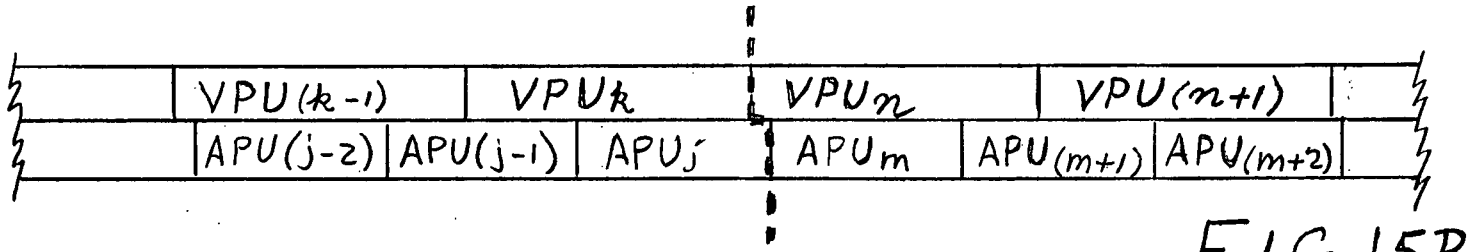

FIG. 15B

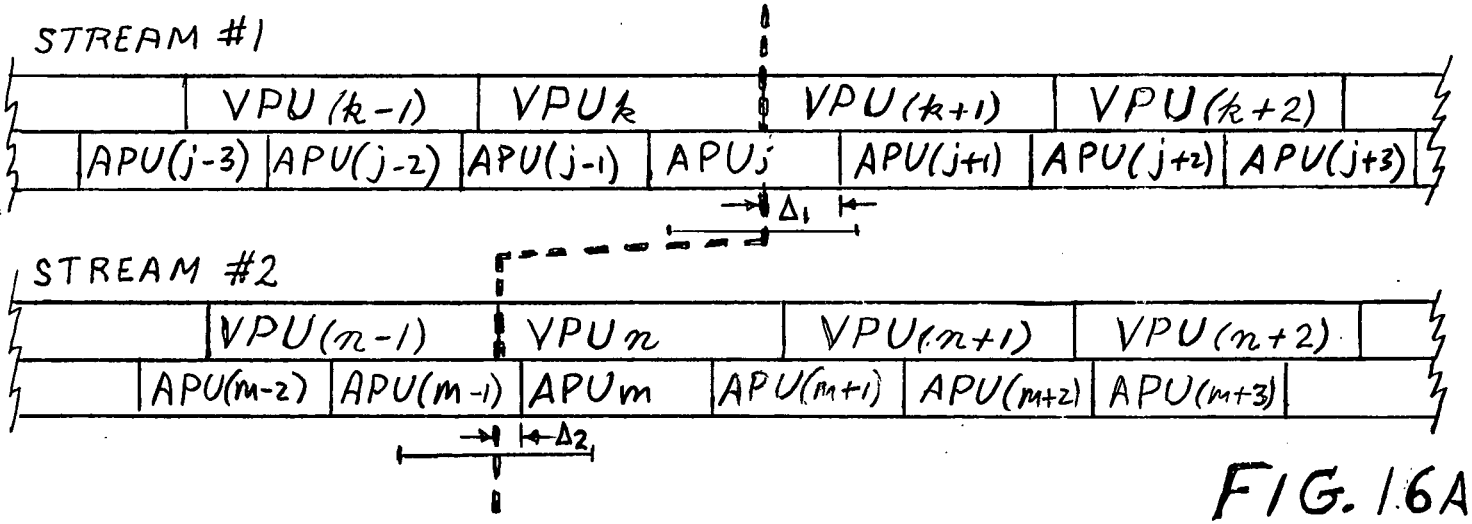

FIG. 16A

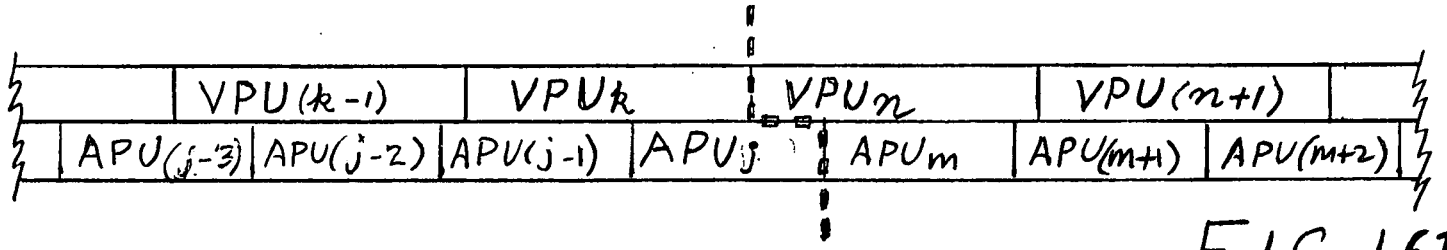

FIG. 16B

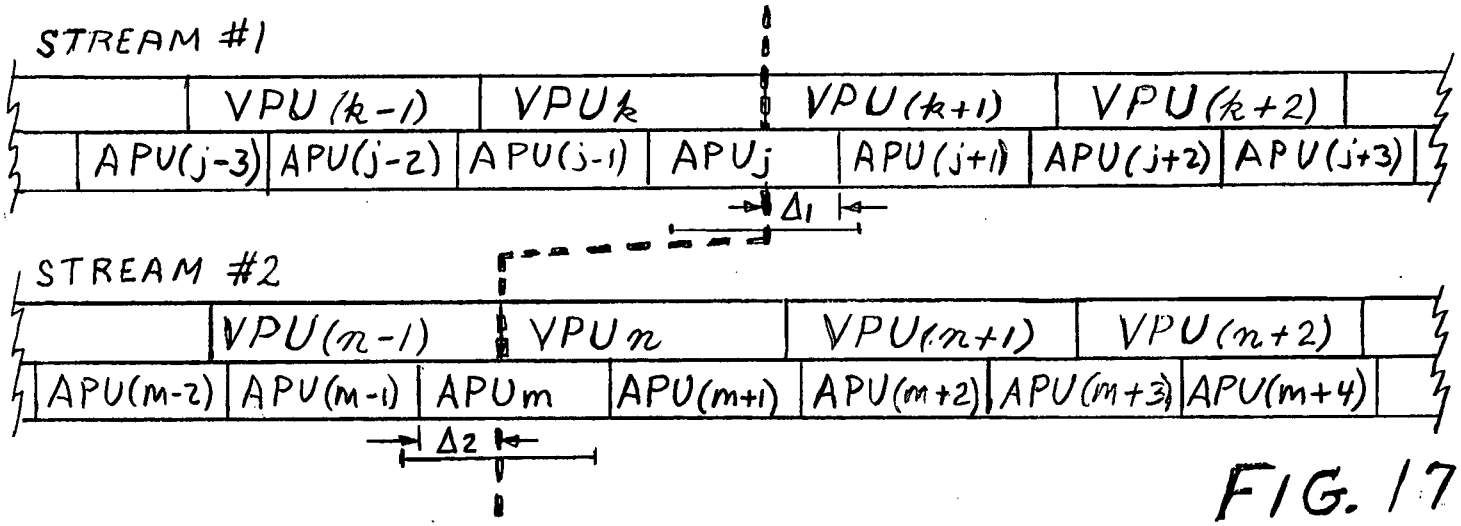

FIG. 17A

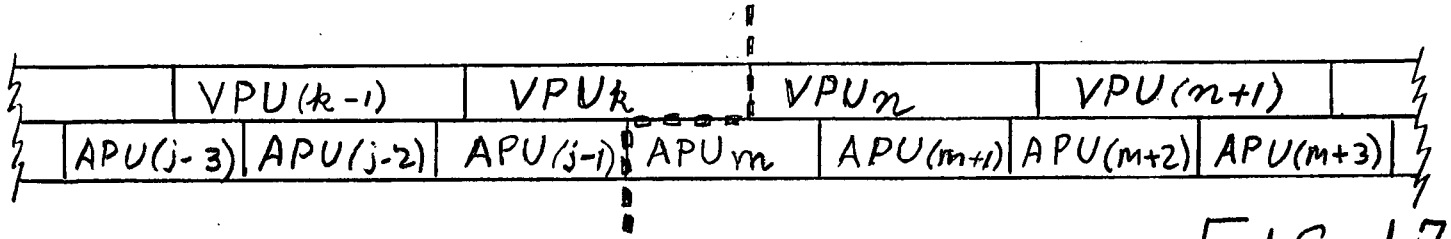

FIG. 17B

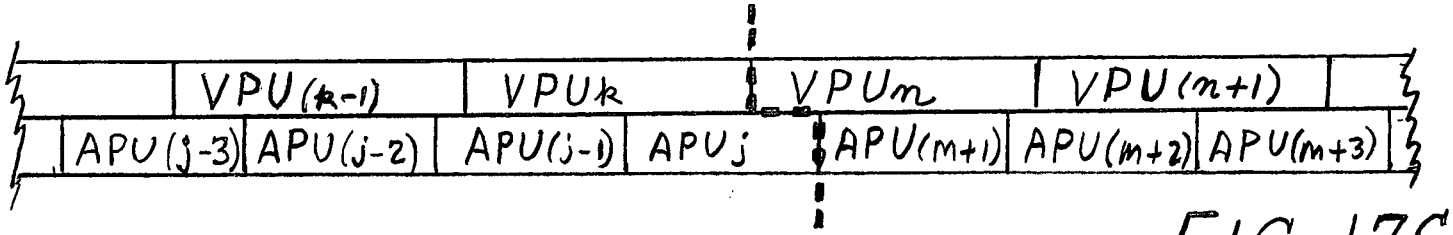

FIG. 17C

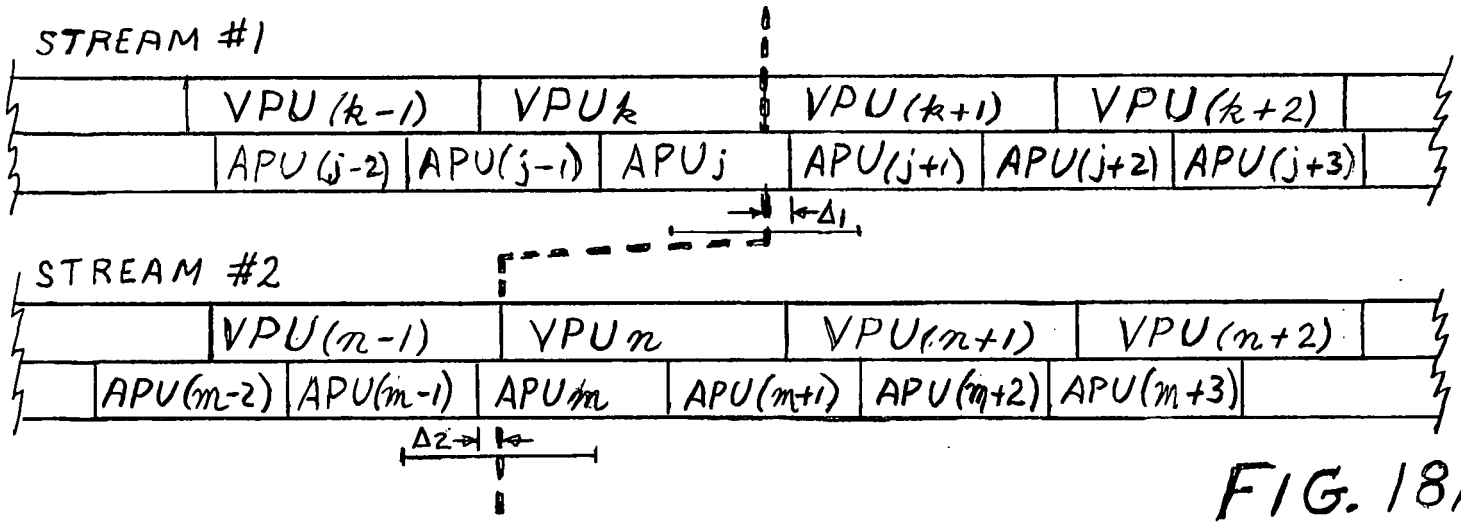

FIG. 18A

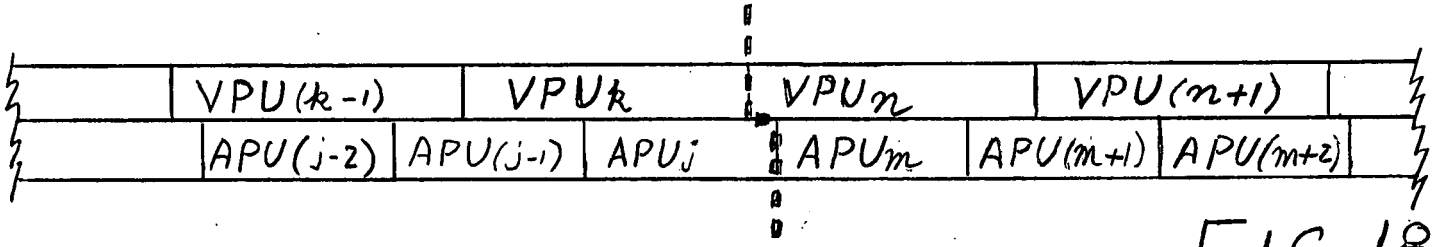

FIG. 18B

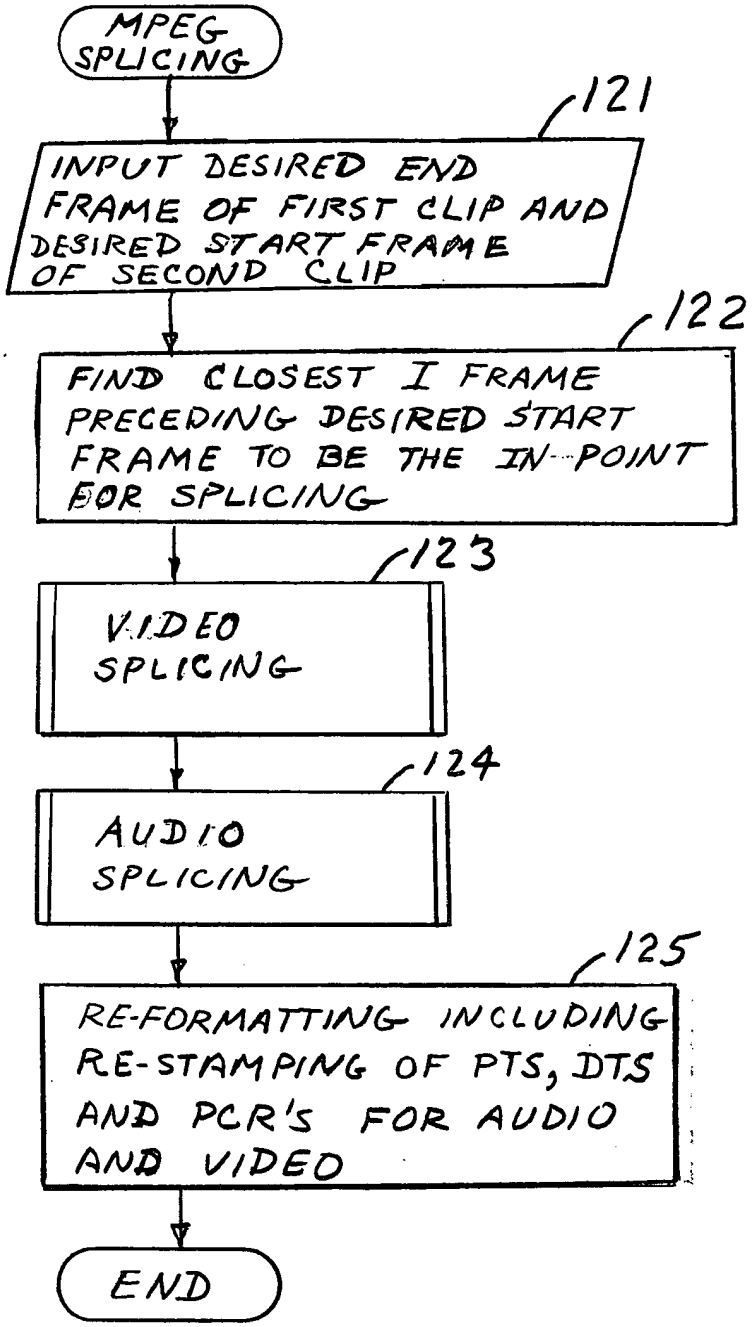
MPEG SPLICING
121
INPUT DESIRED END FRAME OF FIRST CLIP AND DESIRED START FRAME OF SECOND CLIP
122
FIND CLOSEST I FRAME PRECEDING DESIRED START FRAME TO BE THE IN-POINT FOR SPLICING
123
VIDEO SPLICING
124
AUDIO SPLICING
125
RE-FORMATTING INCLUDING RE-STAMPING OF PTS, DTS AND PCR'S FOR AUDIO AND VIDEO
END

FIG. 19

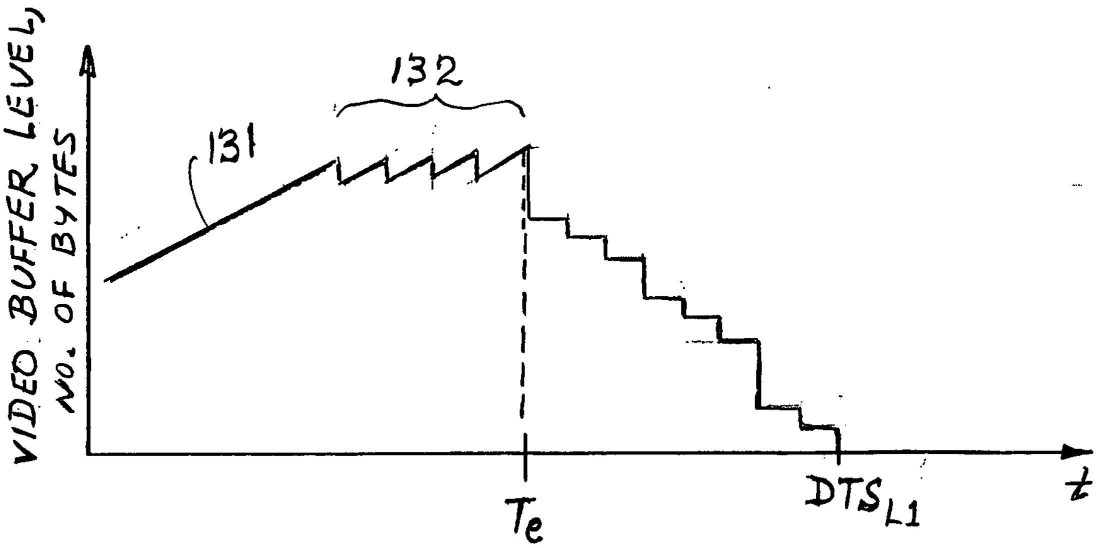

FIG. 20A

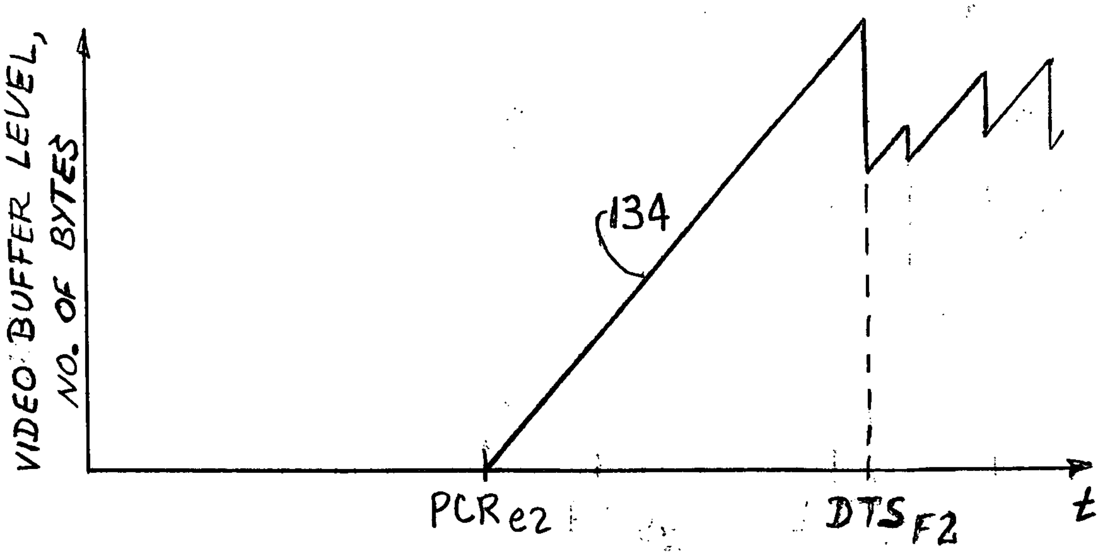

FIG. 20B

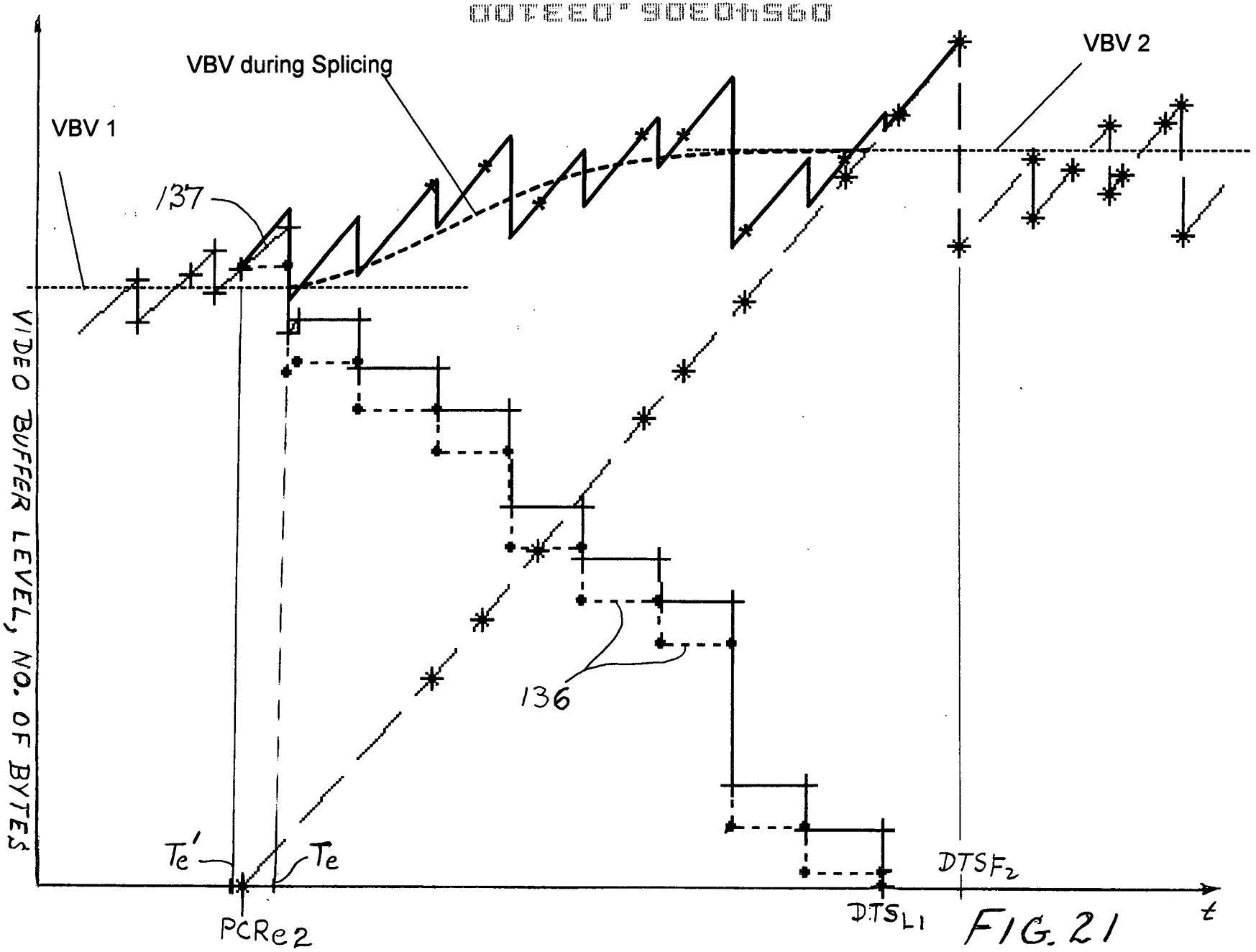

FIG. 21

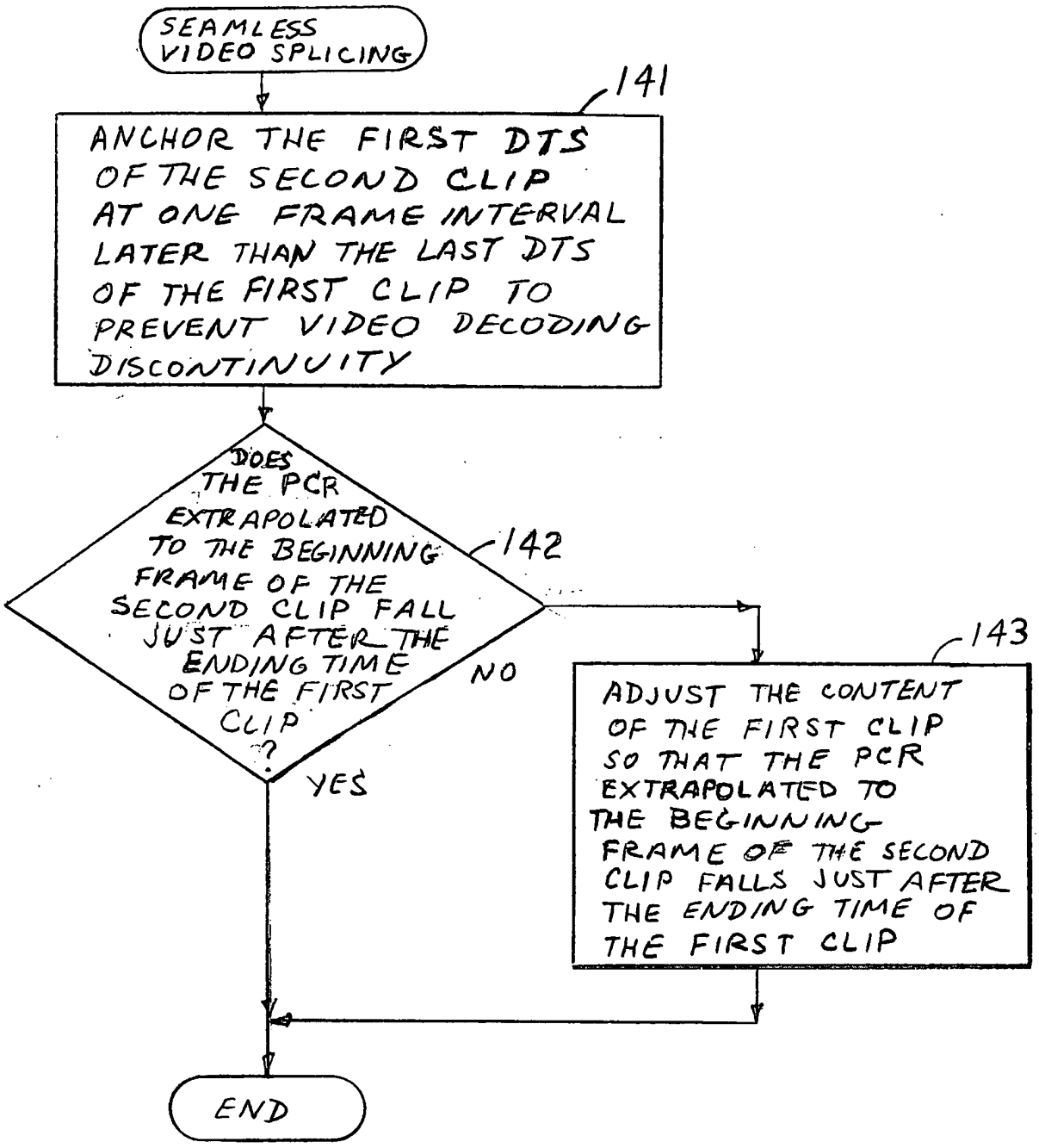
SEAMLESS VIDEO SPLICING
141
ANCHOR THE FIRST DTS OF THE SECOND CLIP AT ONE FRAME INTERVAL LATER THAN THE LAST DTS OF THE FIRST CLIP TO PREVENT VIDEO DECODING DISCONTINUITY
DOES THE PCR EXTRAPOLATED TO THE BEGINNING FRAME OF THE SECOND CLIP FALL JUST AFTER THE ENDING TIME OF THE FIRST CLIP ?
142
NO
YES
143
ADJUST THE CONTENT OF THE FIRST CLIP SO THAT THE PCR EXTRAPOLATED TO THE BEGINNING FRAME OF THE SECOND CLIP FALLS JUST AFTER THE ENDING TIME OF THE FIRST CLIP
END

FIG. 22

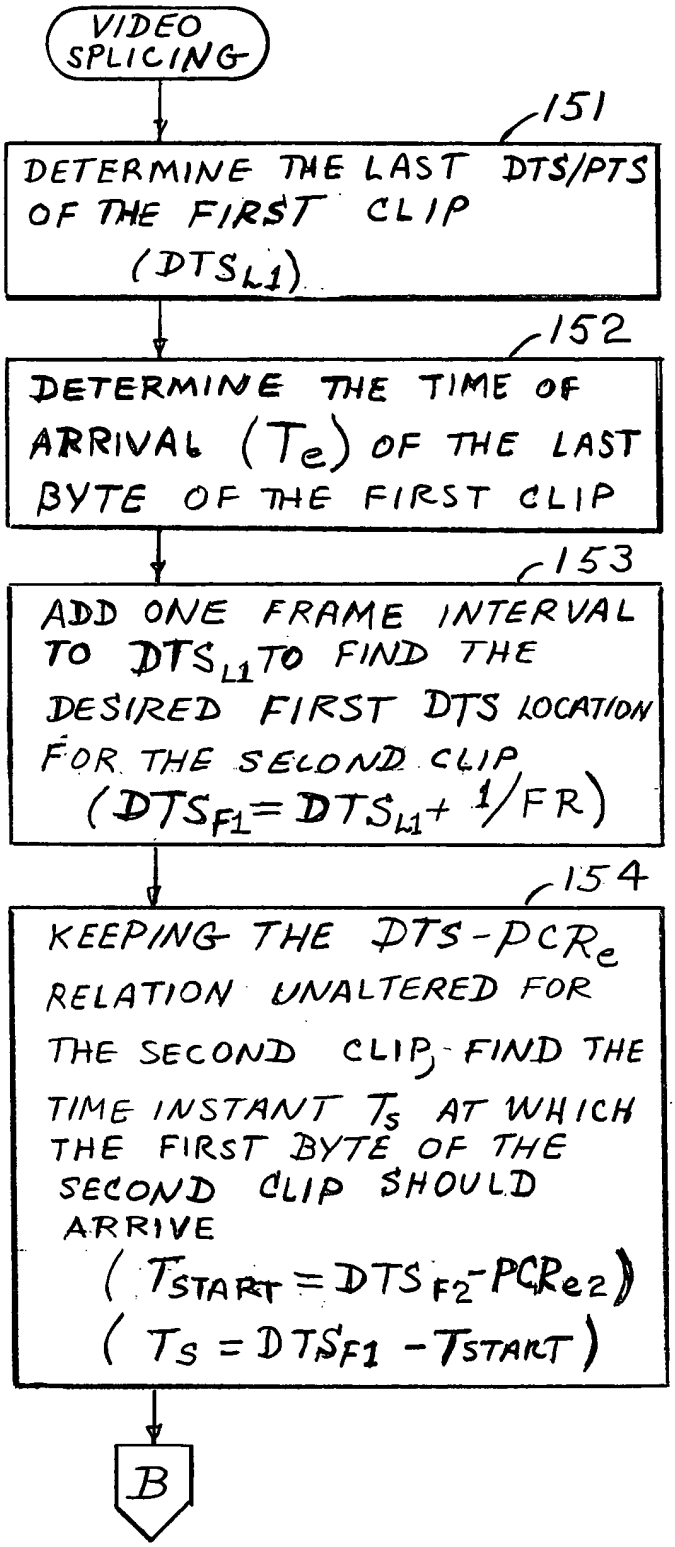
VIDEO SPLICING
151
DETERMINE THE LAST DTS/PTS OF THE FIRST CLIP (DTS_{L1})
152
DETERMINE THE TIME OF ARRIVAL (T_e) OF THE LAST BYTE OF THE FIRST CLIP
153
ADD ONE FRAME INTERVAL TO DTS_{L1} TO FIND THE DESIRED FIRST DTS LOCATION FOR THE SECOND CLIP ($DTS_{F1} = DTS_{L1} + 1/FR$)
154
KEEPING THE DTS-PCR_e RELATION UNALTERED FOR THE SECOND CLIP, FIND THE TIME INSTANT T_s AT WHICH THE FIRST BYTE OF THE SECOND CLIP SHOULD ARRIVE
($T_{START} = DTS_{F2} - PCR_{e2}$)
($T_s = DTS_{F1} - T_{START}$)
B

FIG. 23

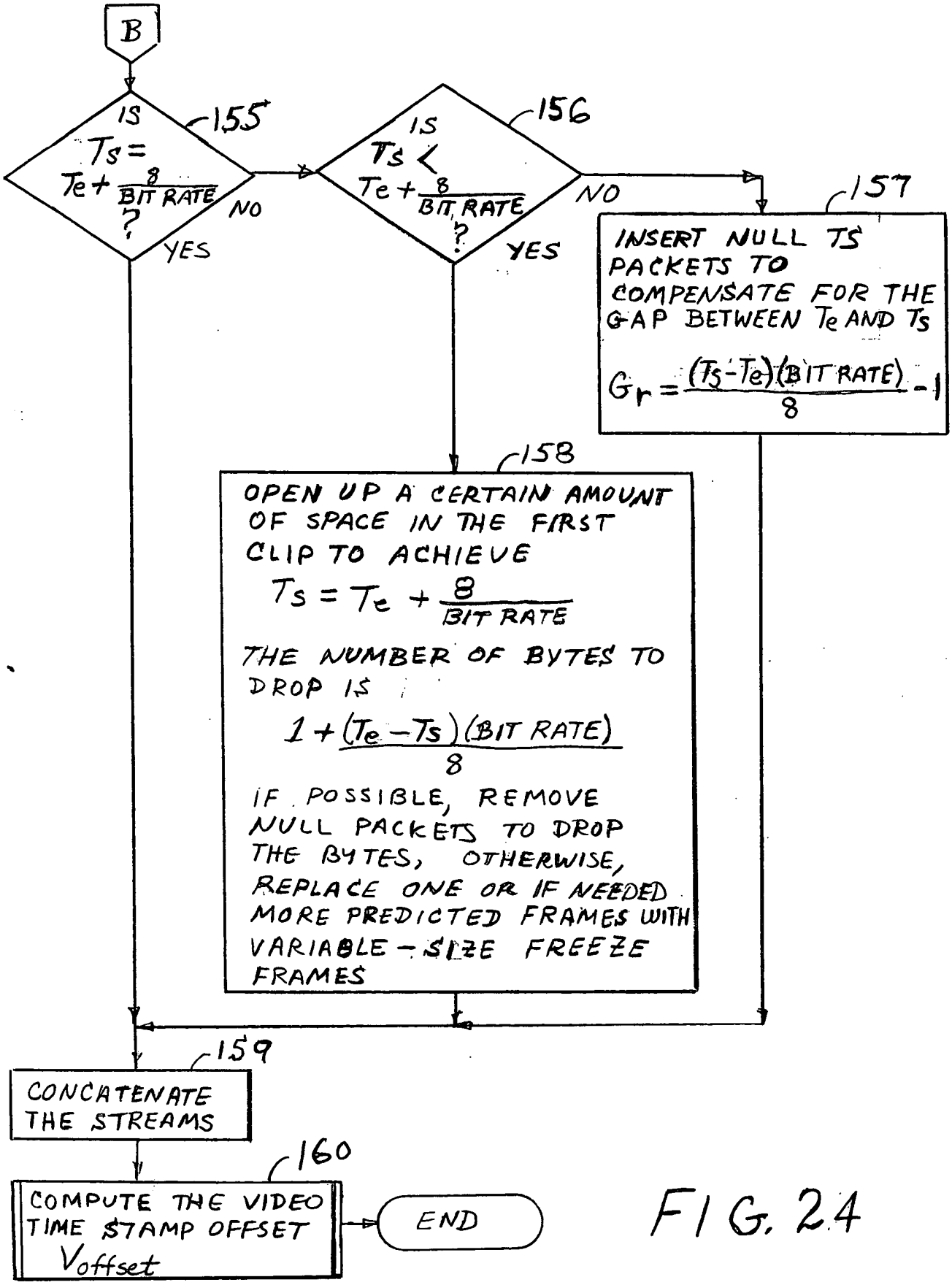
B
IS $T_S = T_e + \frac{8}{BIT\ RATE}$?
155
NO
YES
IS $T_S < T_e + \frac{8}{BIT\ RATE}$?
156
NO
YES
157
INSERT NULL TS PACKETS TO COMPENSATE FOR THE GAP BETWEEN T_e AND T_S
$G_r = \frac{(T_S - T_e)(BIT\ RATE)}{8} - 1$
158
OPEN UP A CERTAIN AMOUNT OF SPACE IN THE FIRST CLIP TO ACHIEVE
$T_S = T_e + \frac{8}{BIT\ RATE}$
THE NUMBER OF BYTES TO DROP IS
$1 + \frac{(T_e - T_S)(BIT\ RATE)}{8}$
IF POSSIBLE, REMOVE NULL PACKETS TO DROP THE BYTES, OTHERWISE, REPLACE ONE OR IF NEEDED MORE PREDICTED FRAMES WITH VARIABLE-SIZE FREEZE FRAMES
159
CONCATENATE THE STREAMS
160
COMPUTE THE VIDEO TIME STAMP OFFSET V_{offset}
END

FIG. 24

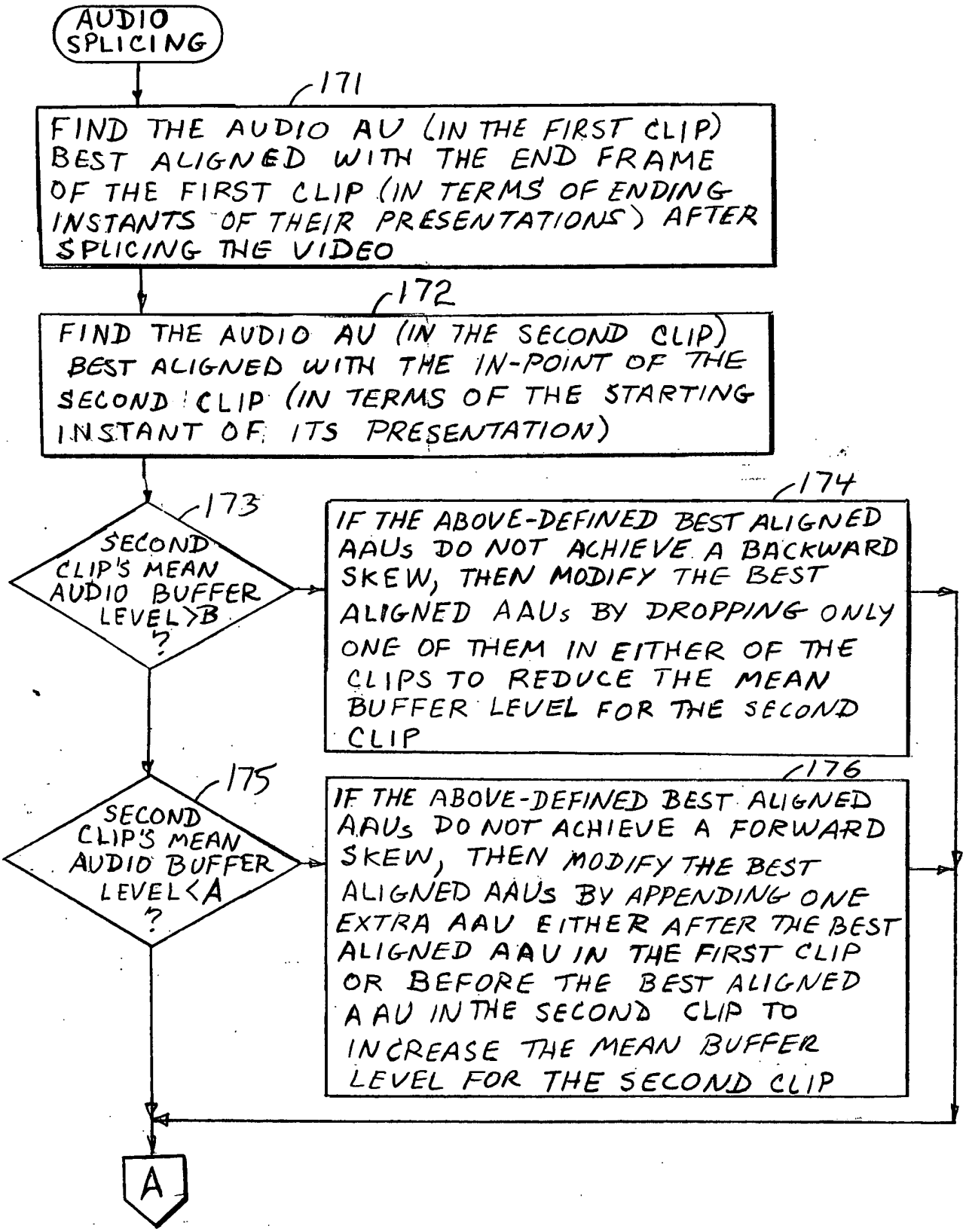
AUDIO SPLICING
171
FIND THE AUDIO AU (IN THE FIRST CLIP) BEST ALIGNED WITH THE END FRAME OF THE FIRST CLIP (IN TERMS OF ENDING INSTANTS OF THEIR PRESENTATIONS) AFTER SPLICING THE VIDEO
172
FIND THE AUDIO AU (IN THE SECOND CLIP) BEST ALIGNED WITH THE IN-POINT OF THE SECOND CLIP (IN TERMS OF THE STARTING INSTANT OF ITS PRESENTATION)
173
SECOND CLIP'S MEAN AUDIO BUFFER LEVEL>B ?
174
IF THE ABOVE-DEFINED BEST ALIGNED AAUs DO NOT ACHIEVE A BACKWARD SKEW, THEN MODIFY THE BEST ALIGNED AAUs BY DROPPING ONLY ONE OF THEM IN EITHER OF THE CLIPS TO REDUCE THE MEAN BUFFER LEVEL FOR THE SECOND CLIP
175
SECOND CLIP'S MEAN AUDIO BUFFER LEVEL<A ?
176
IF THE ABOVE-DEFINED BEST ALIGNED AAUs DO NOT ACHIEVE A FORWARD SKEW, THEN MODIFY THE BEST ALIGNED AAUs BY APPENDING ONE EXTRA AAU EITHER AFTER THE BEST ALIGNED AAU IN THE FIRST CLIP OR BEFORE THE BEST ALIGNED AAU IN THE SECOND CLIP TO INCREASE THE MEAN BUFFER LEVEL FOR THE SECOND CLIP
A

FIG. 25

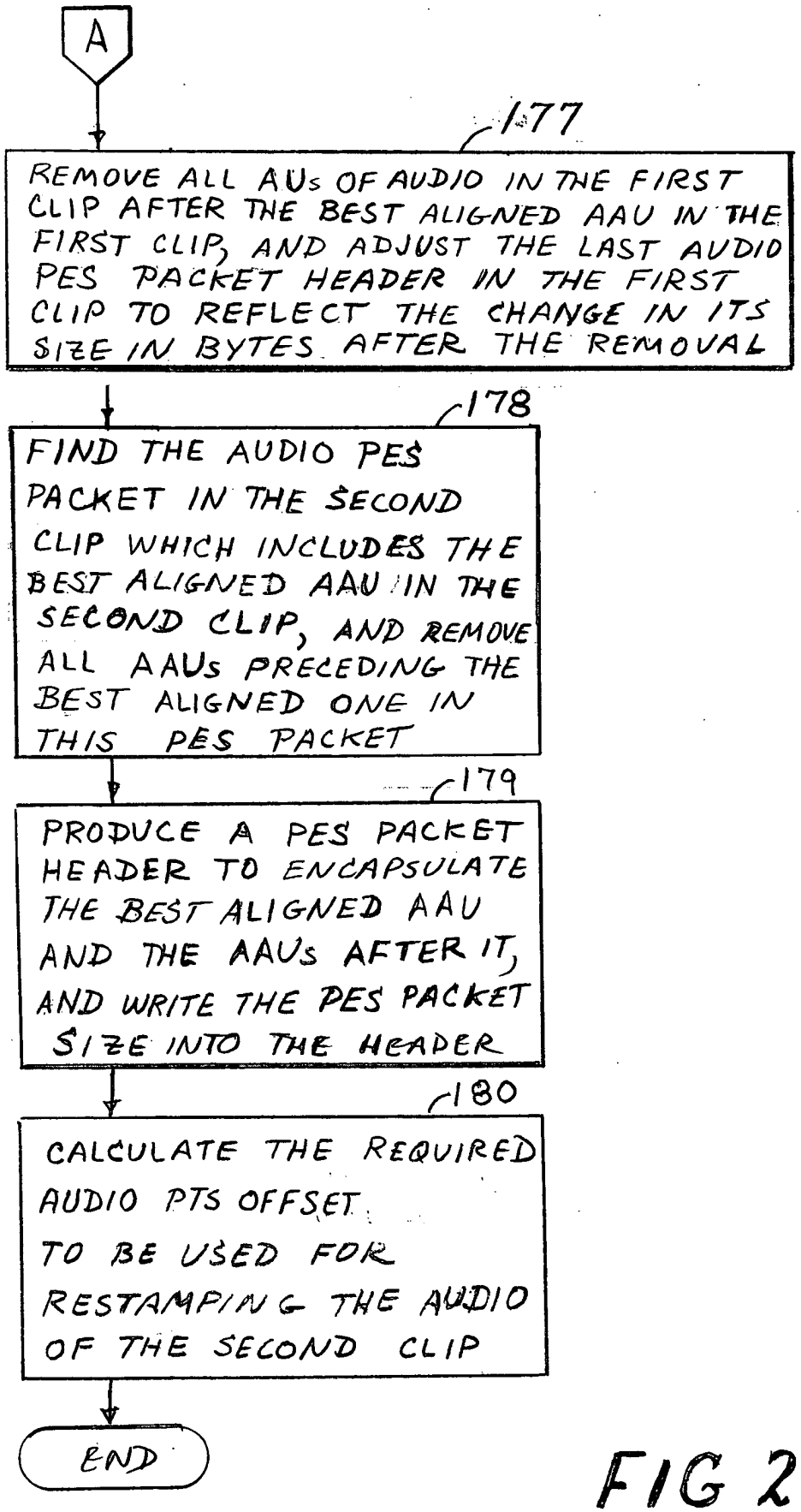
A
177
REMOVE ALL AUs OF AUDIO IN THE FIRST CLIP AFTER THE BEST ALIGNED AAU IN THE FIRST CLIP, AND ADJUST THE LAST AUDIO PES PACKET HEADER IN THE FIRST CLIP TO REFLECT THE CHANGE IN ITS SIZE IN BYTES AFTER THE REMOVAL
178
FIND THE AUDIO PES PACKET IN THE SECOND CLIP WHICH INCLUDES THE BEST ALIGNED AAU IN THE SECOND CLIP, AND REMOVE ALL AAUs PRECEDING THE BEST ALIGNED ONE IN THIS PES PACKET
179
PRODUCE A PES PACKET HEADER TO ENCAPSULATE THE BEST ALIGNED AAU AND THE AAUs AFTER IT, AND WRITE THE PES PACKET SIZE INTO THE HEADER
180
CALCULATE THE REQUIRED AUDIO PTS OFFSET TO BE USED FOR RESTAMPING THE AUDIO OF THE SECOND CLIP
END

FIG 26

CASE	SECOND CLIP HAS A HIGH MEAN AUDIO BUFFER LEVEL	SECOND CLIP HAS A LOW MEAN AUDIO BUFFER LEVEL
FIG. 11A	USE FIG. 28	USE FIG. 11B OR 11C
FIG. 12A	USE FIG. 12B	USE FIG. 29
FIG. 13A	USE FIG. 13B	USE FIG. 30
FIG. 14A	USE FIG. 31	USE FIG. 14B
FIG. 15A	USE FIG. 15B	USE FIG. 32
FIG. 16A	USE FIG. 33	USE FIG. 16B
FIG. 17A	USE FIG. 17B OR 17C	USE FIG. 34
FIG. 18A	USE FIG. 35	USE FIG. 18B

FIG. 27

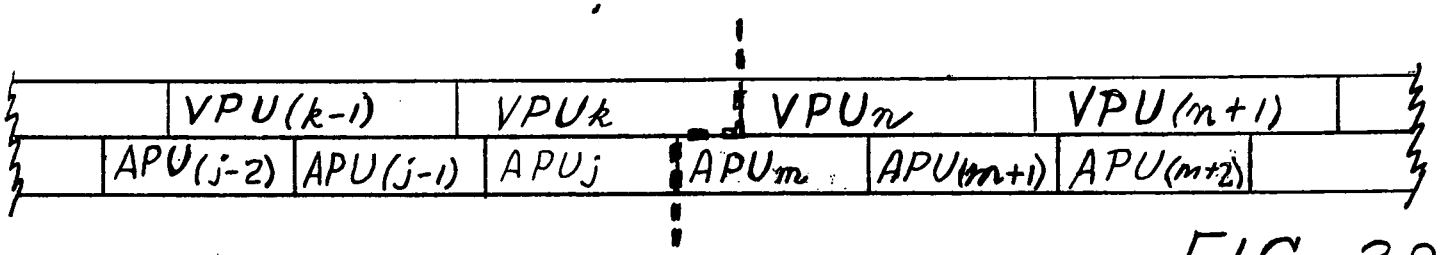

FIG. 28

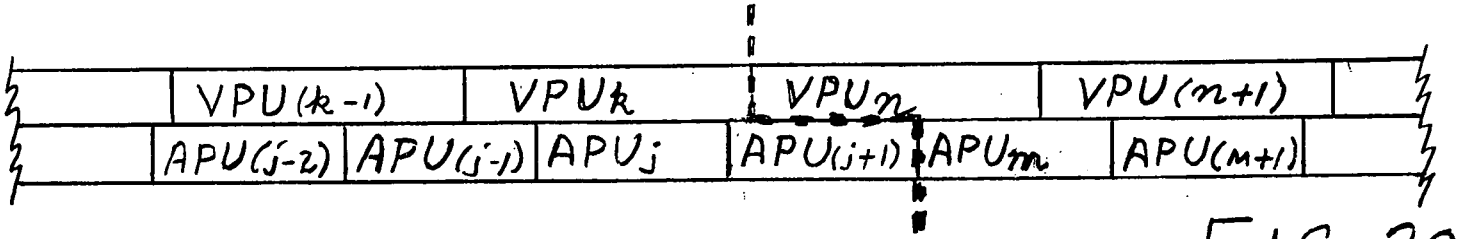

FIG. 29

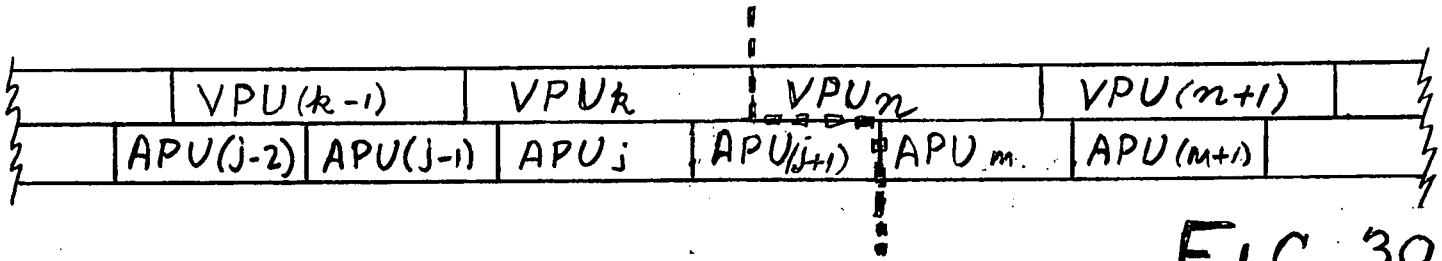

FIG. 30

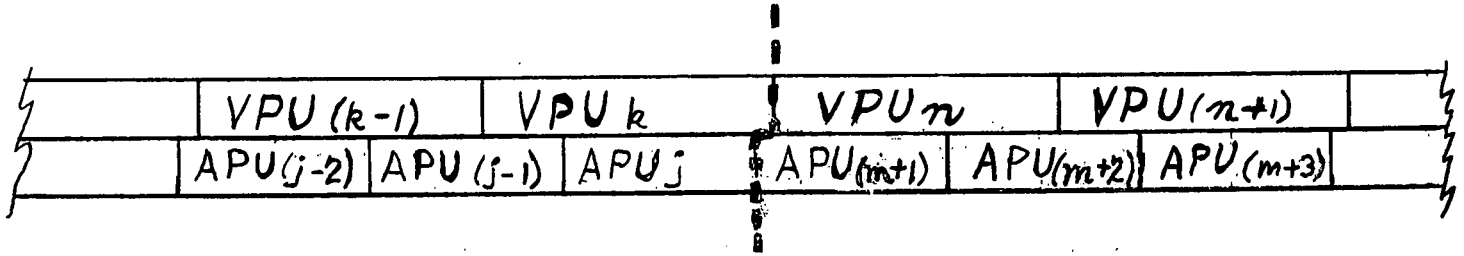

FIG. 31

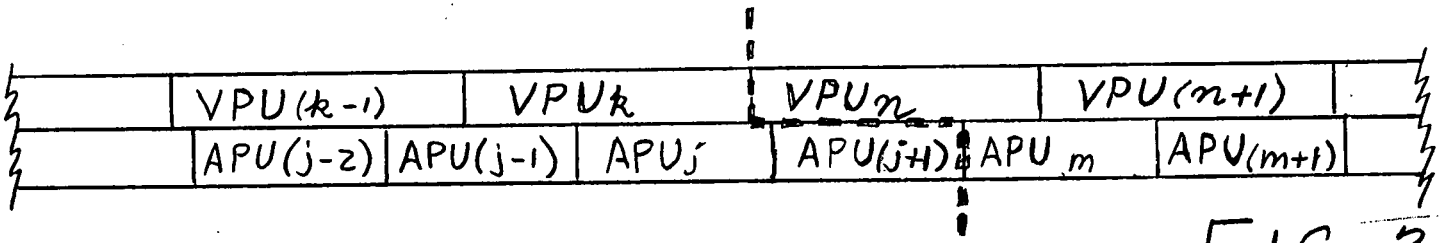

FIG. 32

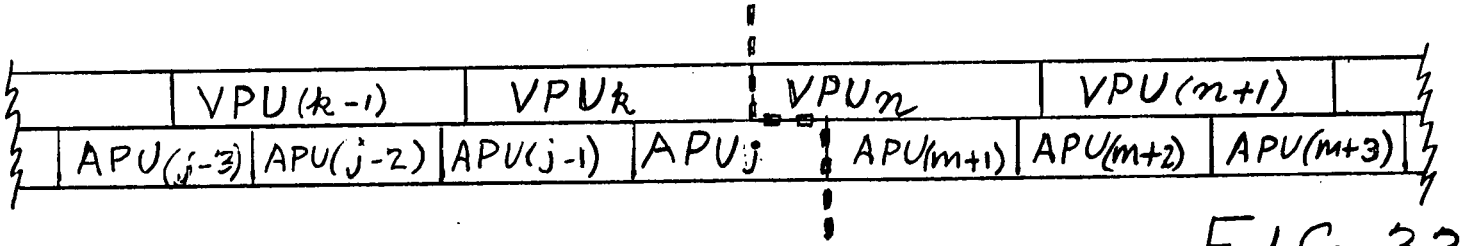

FIG. 33

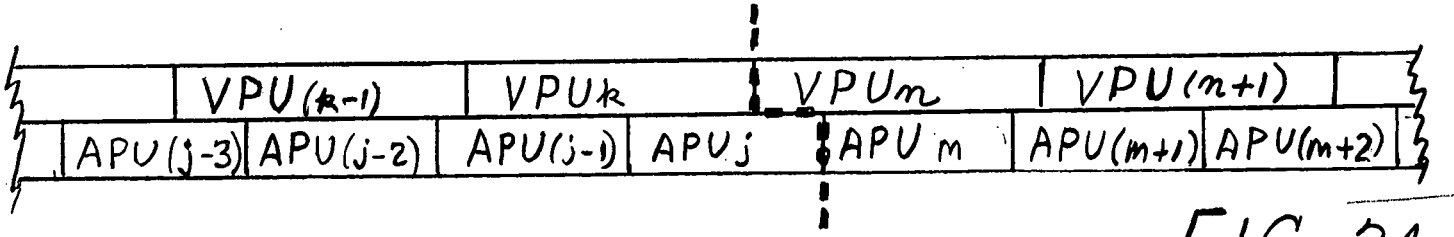

FIG. 34

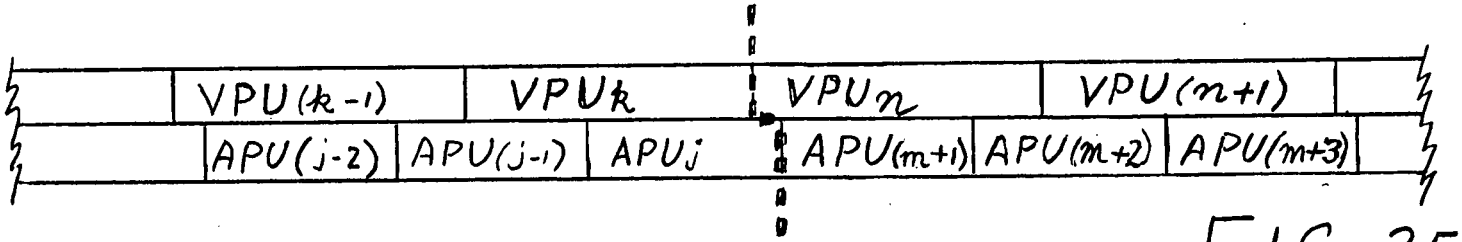

FIG. 35

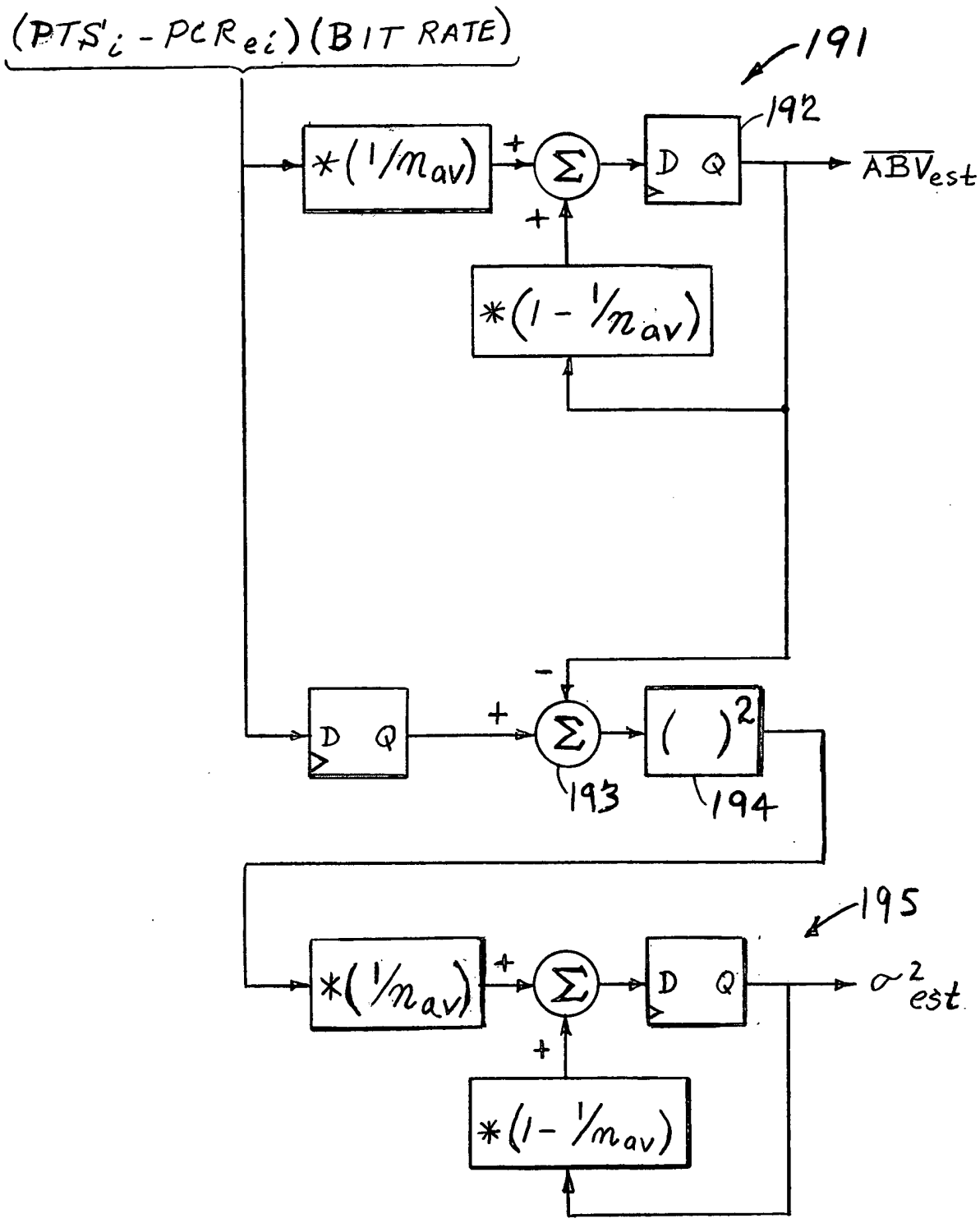
(PTS'i - PCRei)(BIT RATE)
191
192
*(1/nav)
+
Σ
D Q
ABVest
+
*(1 - 1/nav)
D Q
+
Σ
193
( )2
194
195
*(1/nav)
+
Σ
D Q
σ2est
+
*(1 - 1/nav)

FIG. 36

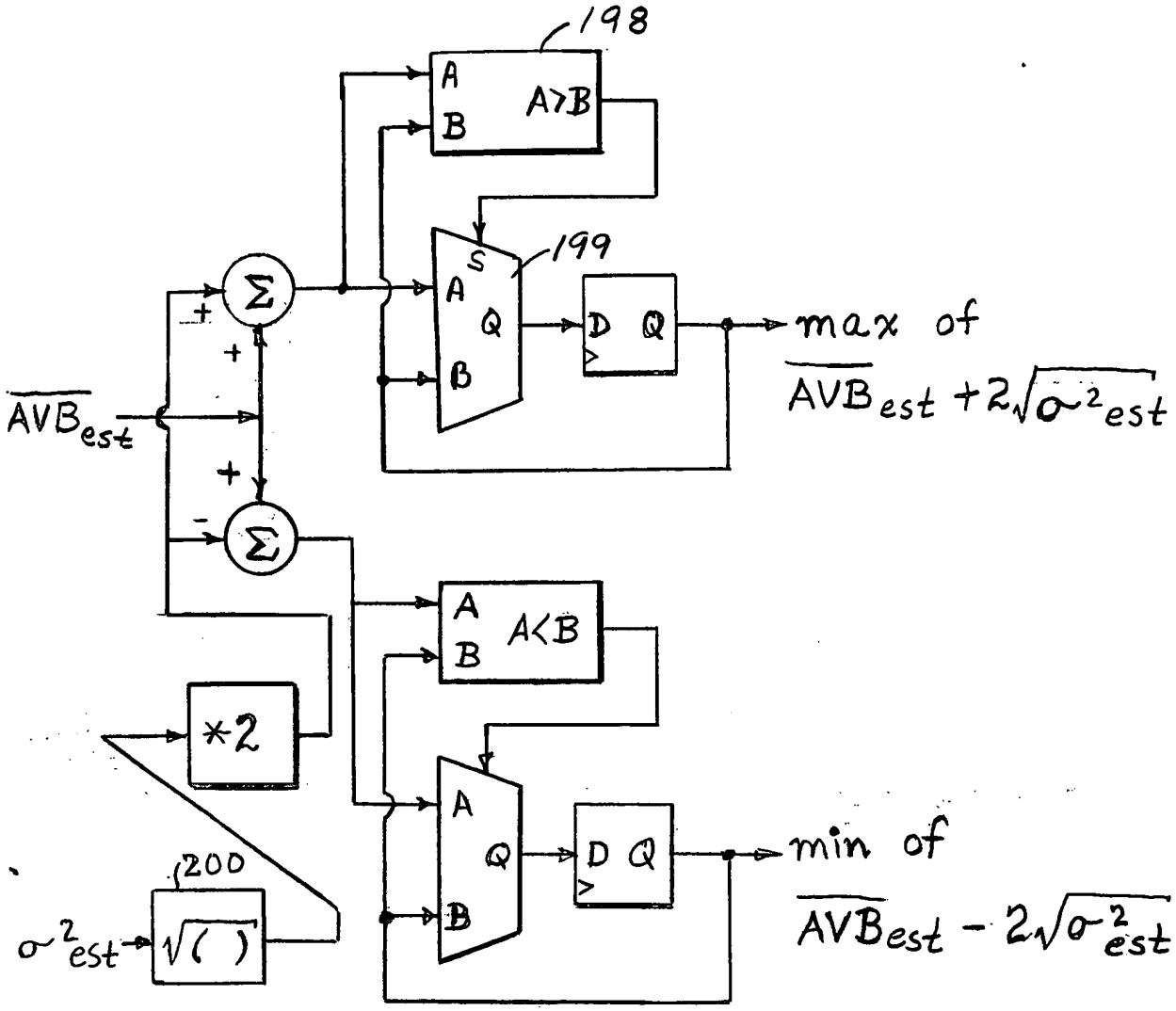
198
A
B
A>B
199
S
A
Q
B
D Q
max of
$\overline{AVB}_{est} + 2\sqrt{\sigma^2_{est}}$
$\overline{AVB}_{est}$
+
+
+
-
Σ
Σ
A
B
A<B
*2
A
Q
B
D Q
min of
$\overline{AVB}_{est} - 2\sqrt{\sigma^2_{est}}$
200
σ^2_{est}
√( )

FIG. 37

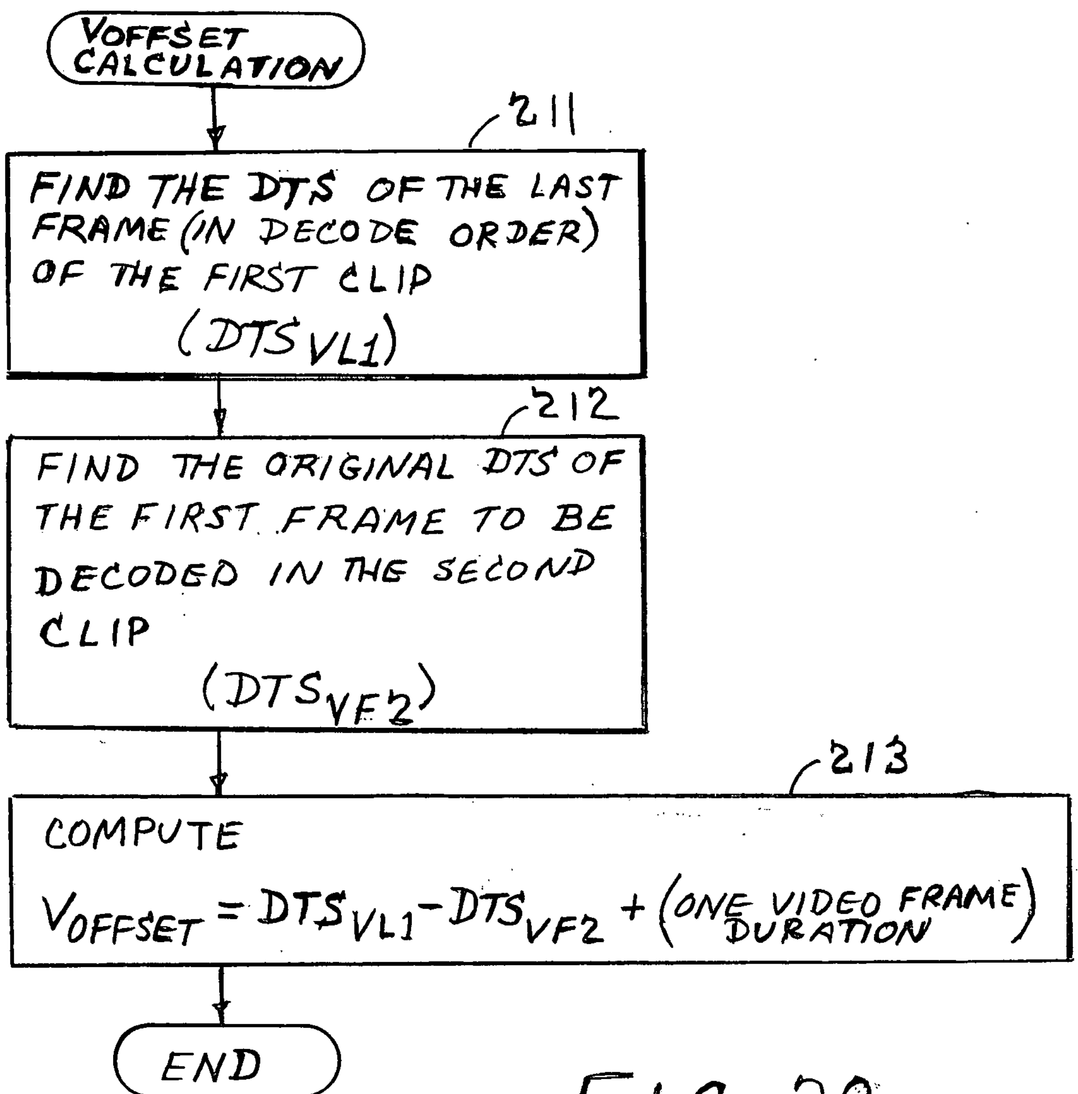
VOFFSET CALCULATION
211
FIND THE DTS OF THE LAST FRAME (IN DECODE ORDER) OF THE FIRST CLIP
(DTS_{VL1})
212
FIND THE ORIGINAL DTS OF THE FIRST FRAME TO BE DECODED IN THE SECOND CLIP
(DTS_{VF2})
213
COMPUTE
$V_{OFFSET} = DTS_{VL1} - DTS_{VF2} + (\text{ONE VIDEO FRAME DURATION})$
END

FIG. 38

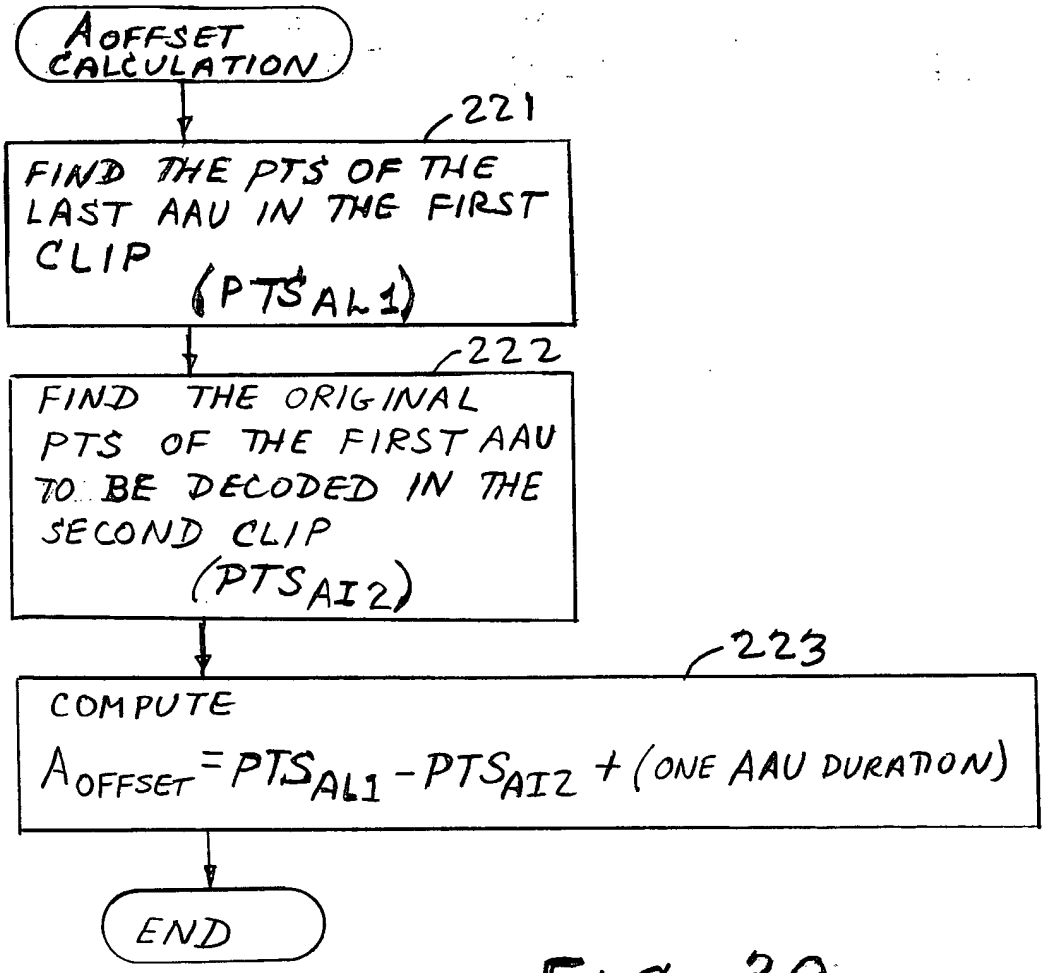
A OFFSET CALCULATION
221
FIND THE PTS OF THE LAST AAU IN THE FIRST CLIP
(PTSAL1)
222
FIND THE ORIGINAL PTS OF THE FIRST AAU TO BE DECODED IN THE SECOND CLIP
(PTSAI2)
223
COMPUTE
AOFFSET = PTSAL1 - PTSAI2 + (ONE AAU DURATION)
END

FIG. 39

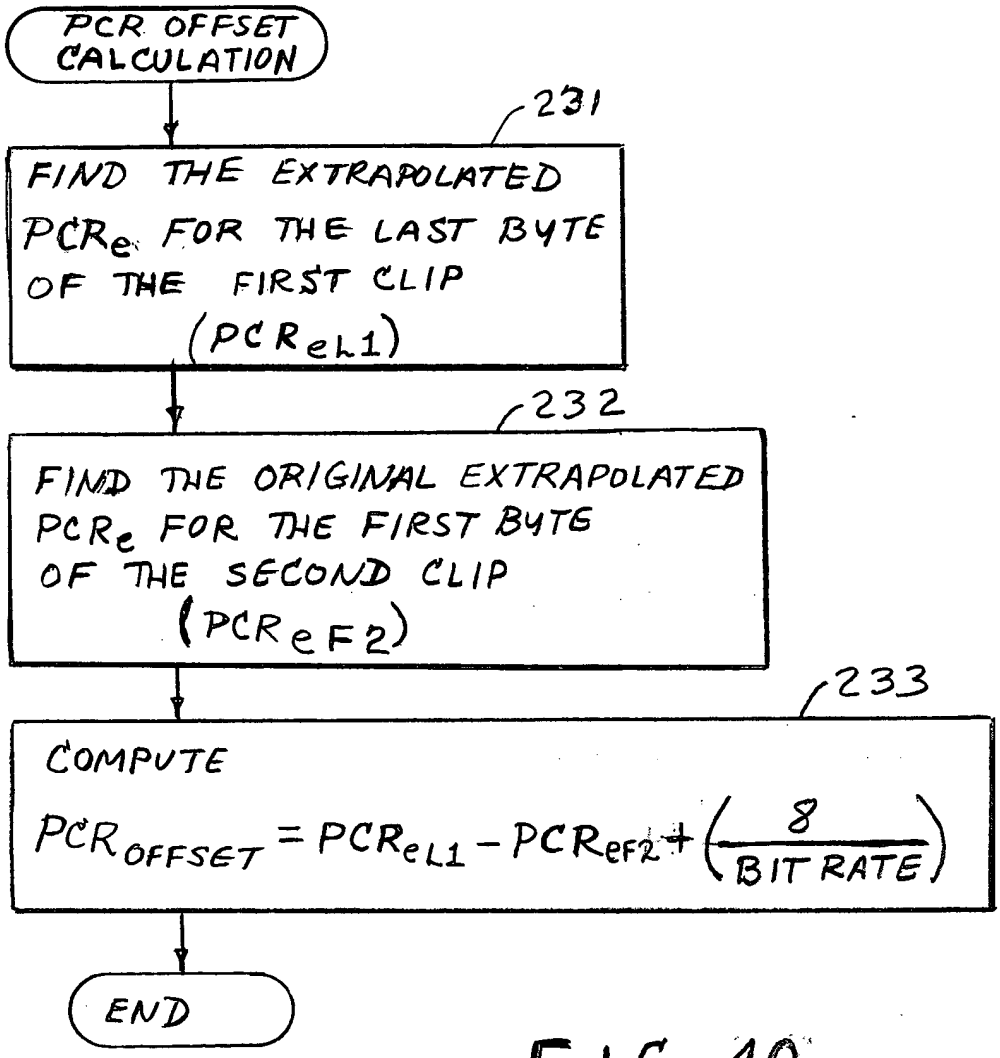
PCR OFFSET CALCULATION
231
FIND THE EXTRAPOLATED PCRe FOR THE LAST BYTE OF THE FIRST CLIP (PCReL1)
232
FIND THE ORIGINAL EXTRAPOLATED PCRe FOR THE FIRST BYTE OF THE SECOND CLIP (PCReF2)
233
COMPUTE
PCROFFSET = PCReL1 − PCReF2 + (8 / BIT RATE)
END

FIG. 40

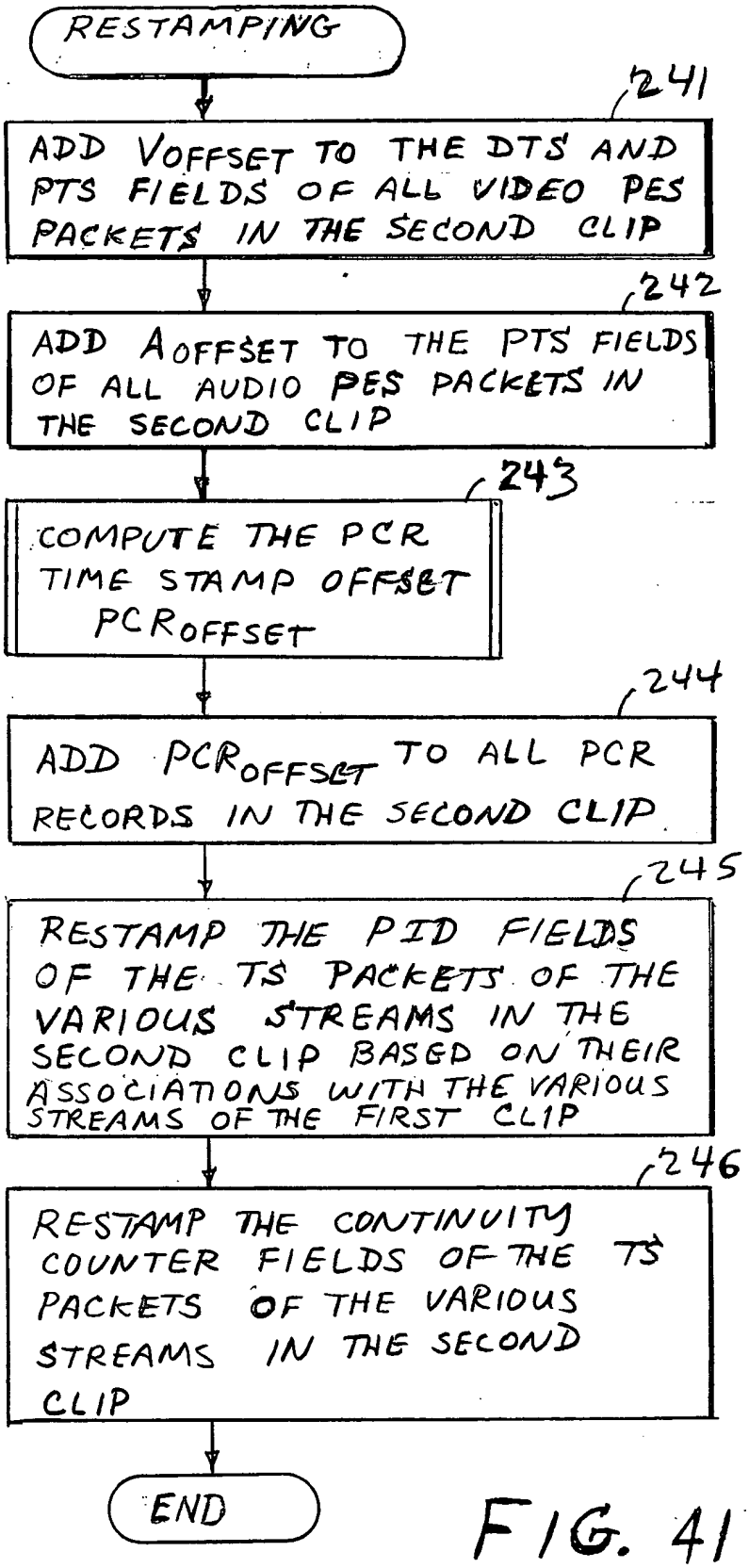
RESTAMPING
241
ADD VOFFSET TO THE DTS AND PTS FIELDS OF ALL VIDEO PES PACKETS IN THE SECOND CLIP
242
ADD AOFFSET TO THE PTS FIELDS OF ALL AUDIO PES PACKETS IN THE SECOND CLIP
243
COMPUTE THE PCR TIME STAMP OFFSET PCROFFSET
244
ADD PCROFFSET TO ALL PCR RECORDS IN THE SECOND CLIP
245
RESTAMP THE PID FIELDS OF THE TS PACKETS OF THE VARIOUS STREAMS IN THE SECOND CLIP BASED ON THEIR ASSOCIATIONS WITH THE VARIOUS STREAMS OF THE FIRST CLIP
246
RESTAMP THE CONTINUITY COUNTER FIELDS OF THE TS PACKETS OF THE VARIOUS STREAMS IN THE SECOND CLIP
END

FIG. 41

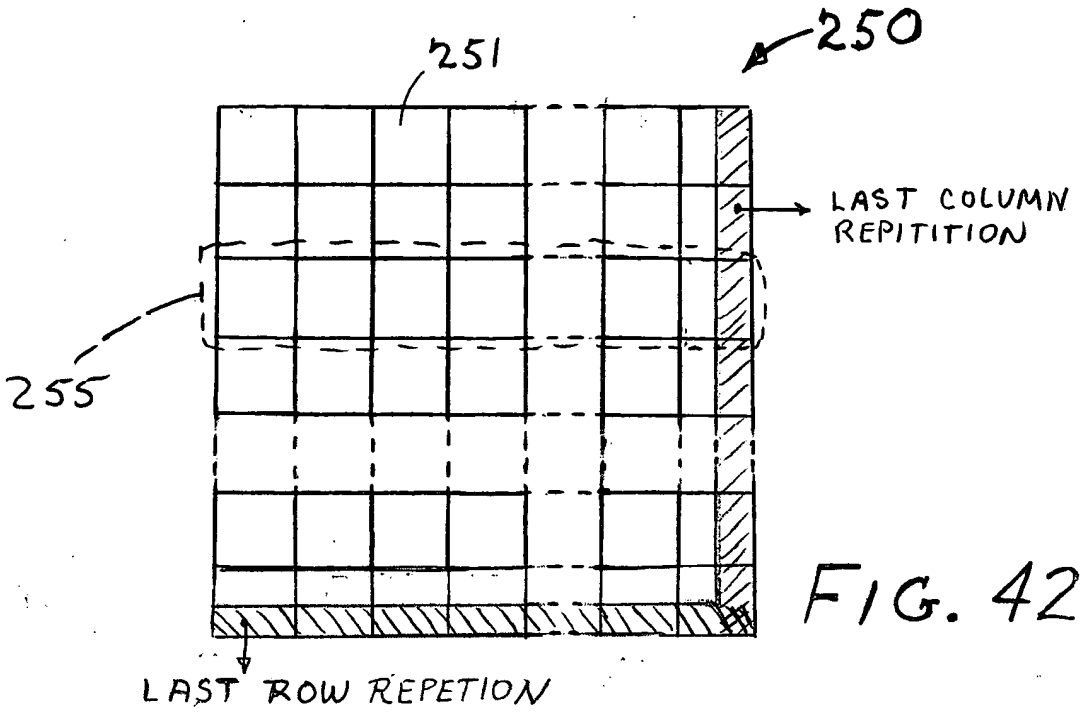
250
251
LAST COLUMN REPITITION
255
LAST ROW REPETION

FIG. 42

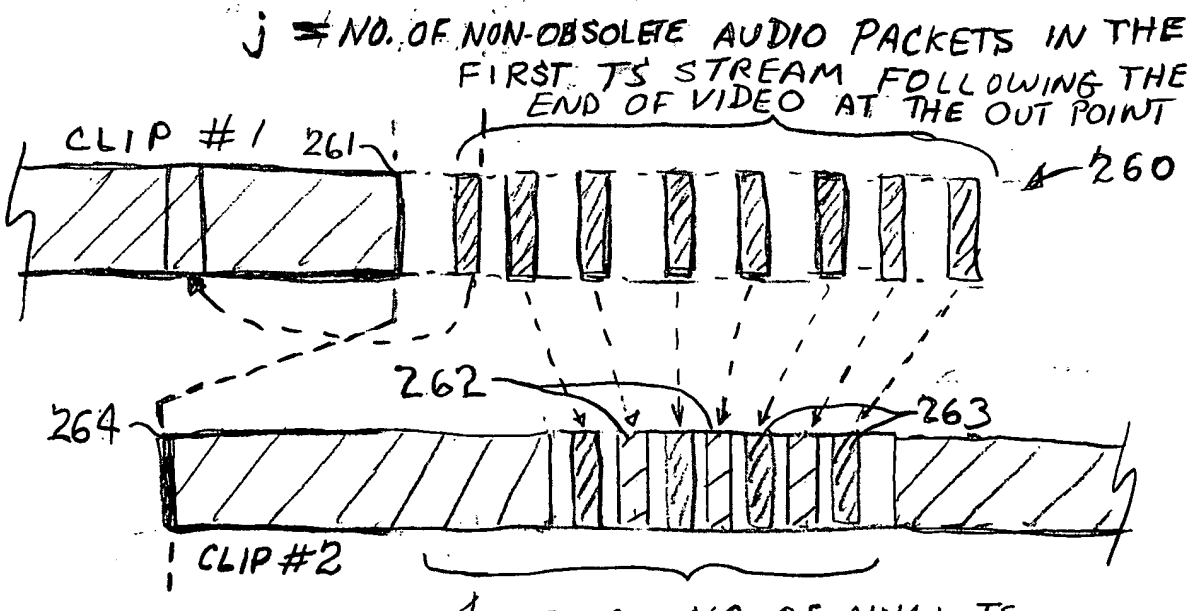
j = NO. OF NON-OBSOLETE AUDIO PACKETS IN THE FIRST TS STREAM FOLLOWING THE END OF VIDEO AT THE OUT POINT
CLIP #1
261
260
262
263
264
CLIP #2
k = TOTAL NO. OF NULL TS PACKETS AND OBSOLETE AUDIO PACKETS IN THE SECOND TS STREAM FOLLOWING THE BEGINNING OF VIDEO AT THE IN POINT

FIG. 43

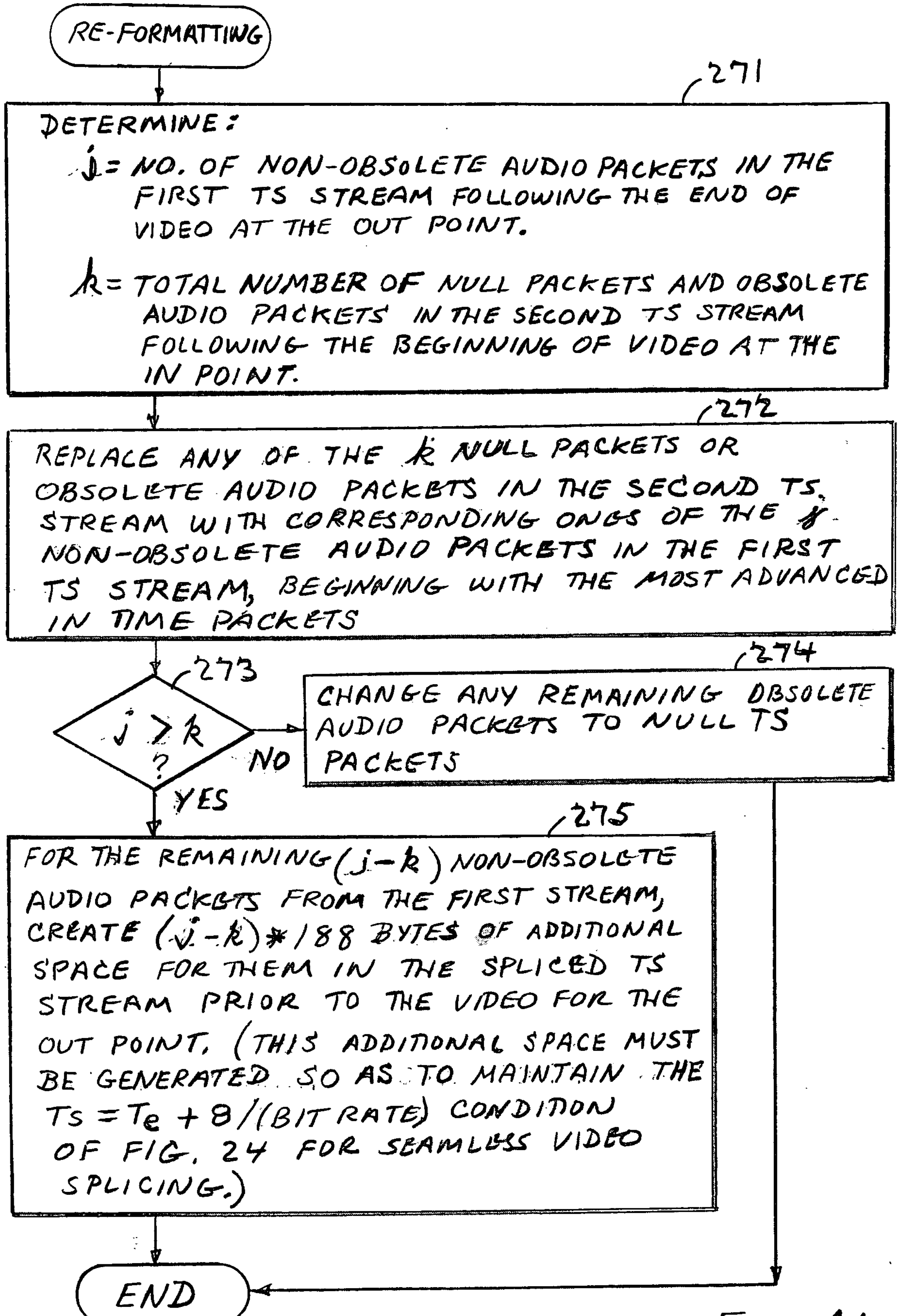
RE-FORMATTING
271
DETERMINE:
j = NO. OF NON-OBSOLETE AUDIO PACKETS IN THE FIRST TS STREAM FOLLOWING THE END OF VIDEO AT THE OUT POINT.
k = TOTAL NUMBER OF NULL PACKETS AND OBSOLETE AUDIO PACKETS IN THE SECOND TS STREAM FOLLOWING THE BEGINNING OF VIDEO AT THE IN POINT.
272
REPLACE ANY OF THE k NULL PACKETS OR OBSOLETE AUDIO PACKETS IN THE SECOND TS STREAM WITH CORRESPONDING ONES OF THE j NON-OBSOLETE AUDIO PACKETS IN THE FIRST TS STREAM, BEGINNING WITH THE MOST ADVANCED IN TIME PACKETS
273
j > k ?
NO
YES
274
CHANGE ANY REMAINING OBSOLETE AUDIO PACKETS TO NULL TS PACKETS
275
FOR THE REMAINING (j−k) NON-OBSOLETE AUDIO PACKETS FROM THE FIRST STREAM, CREATE (j−k)*188 BYTES OF ADDITIONAL SPACE FOR THEM IN THE SPLICED TS STREAM PRIOR TO THE VIDEO FOR THE OUT POINT, (THIS ADDITIONAL SPACE MUST BE GENERATED SO AS TO MAINTAIN THE Ts = Te + 8/(BIT RATE) CONDITION OF FIG. 24 FOR SEAMLESS VIDEO SPLICING.)
END

FIG. 44

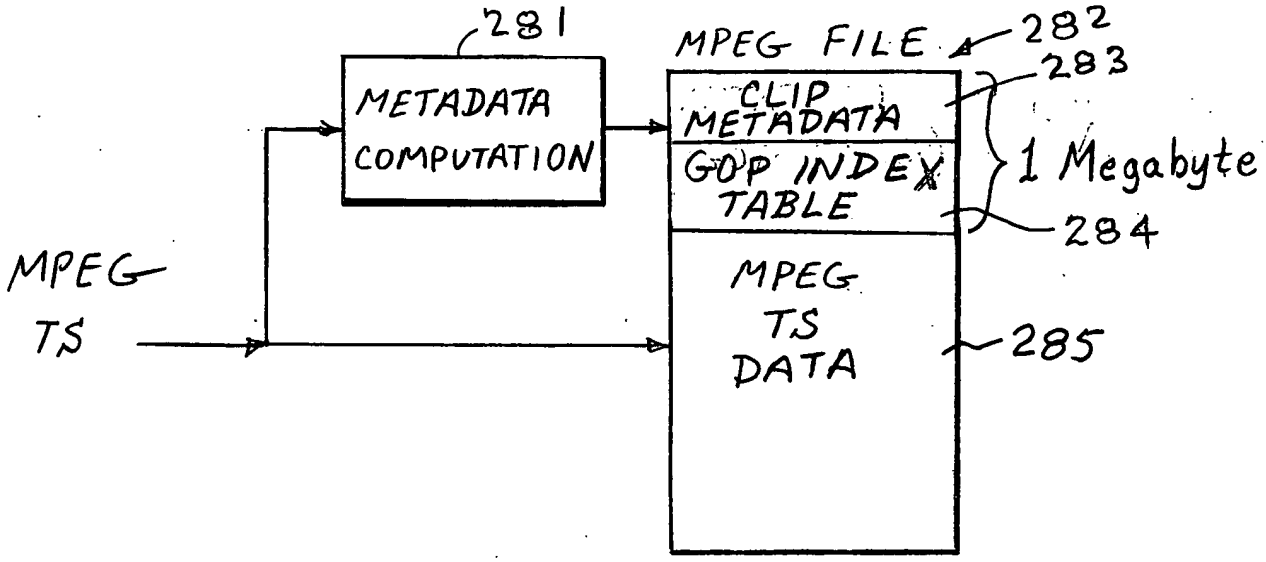

FIG. 45

284

	FRAME NO.	POINTER TO MPEG TS DATA	FLAGS	DTS, PCRe, AND OTHER GOP ATTRIBUTES
GOP 0				
GOP 1				
GOP 2				
GOP 3				
GOP 4				
GOP 5				
⋮				
GOP n				

FIG. 46

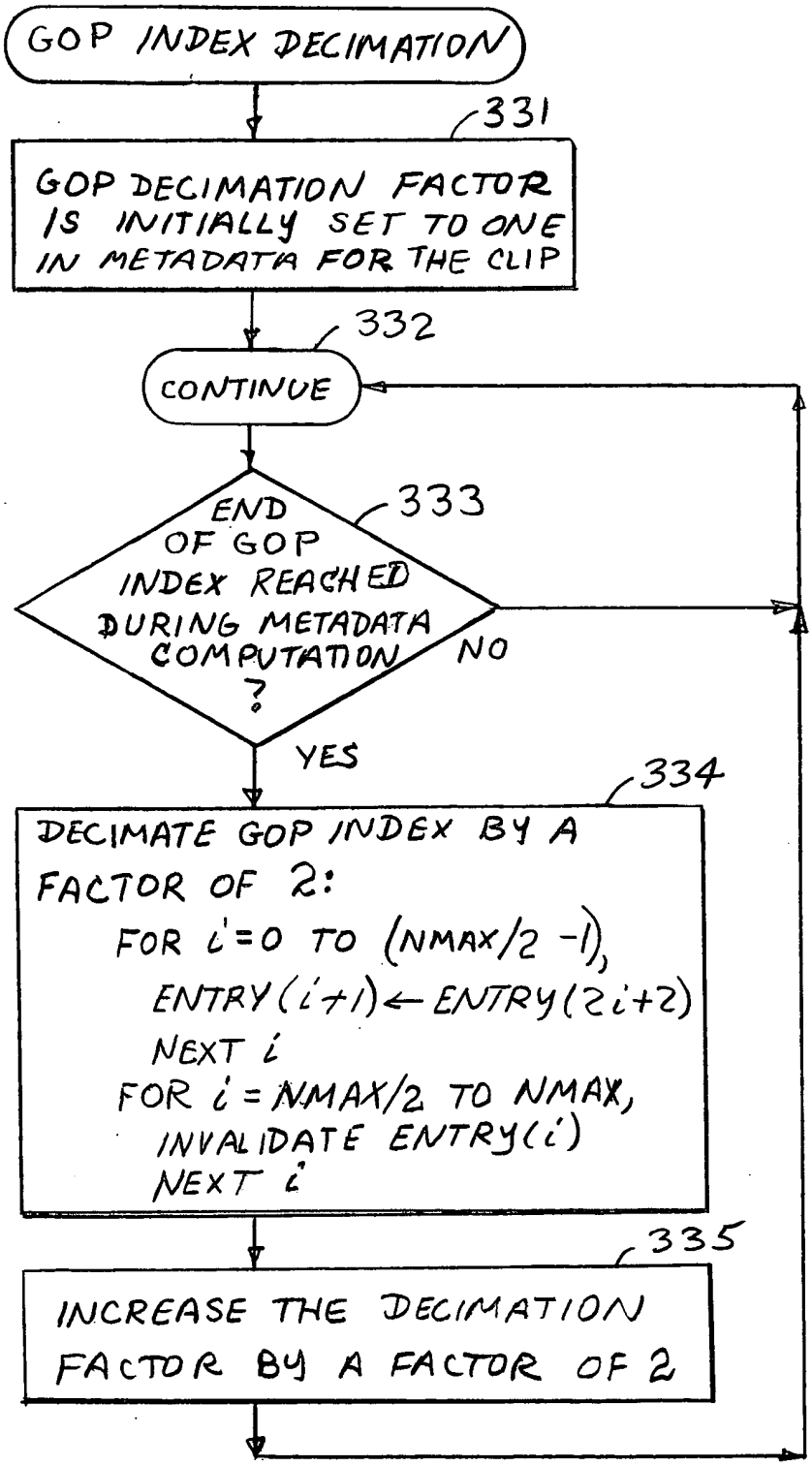
GOP INDEX DECIMATION
331
GOP DECIMATION FACTOR IS INITIALLY SET TO ONE IN METADATA FOR THE CLIP
332
CONTINUE
333
END OF GOP INDEX REACHED DURING METADATA COMPUTATION ?
NO
YES
334
DECIMATE GOP INDEX BY A FACTOR OF 2:
FOR i=0 TO (NMAX/2 -1),
ENTRY(i+1)←ENTRY(2i+2)
NEXT i
FOR i = NMAX/2 TO NMAX,
INVALIDATE ENTRY(i)
NEXT i
335
INCREASE THE DECIMATION FACTOR BY A FACTOR OF 2

FIG. 47

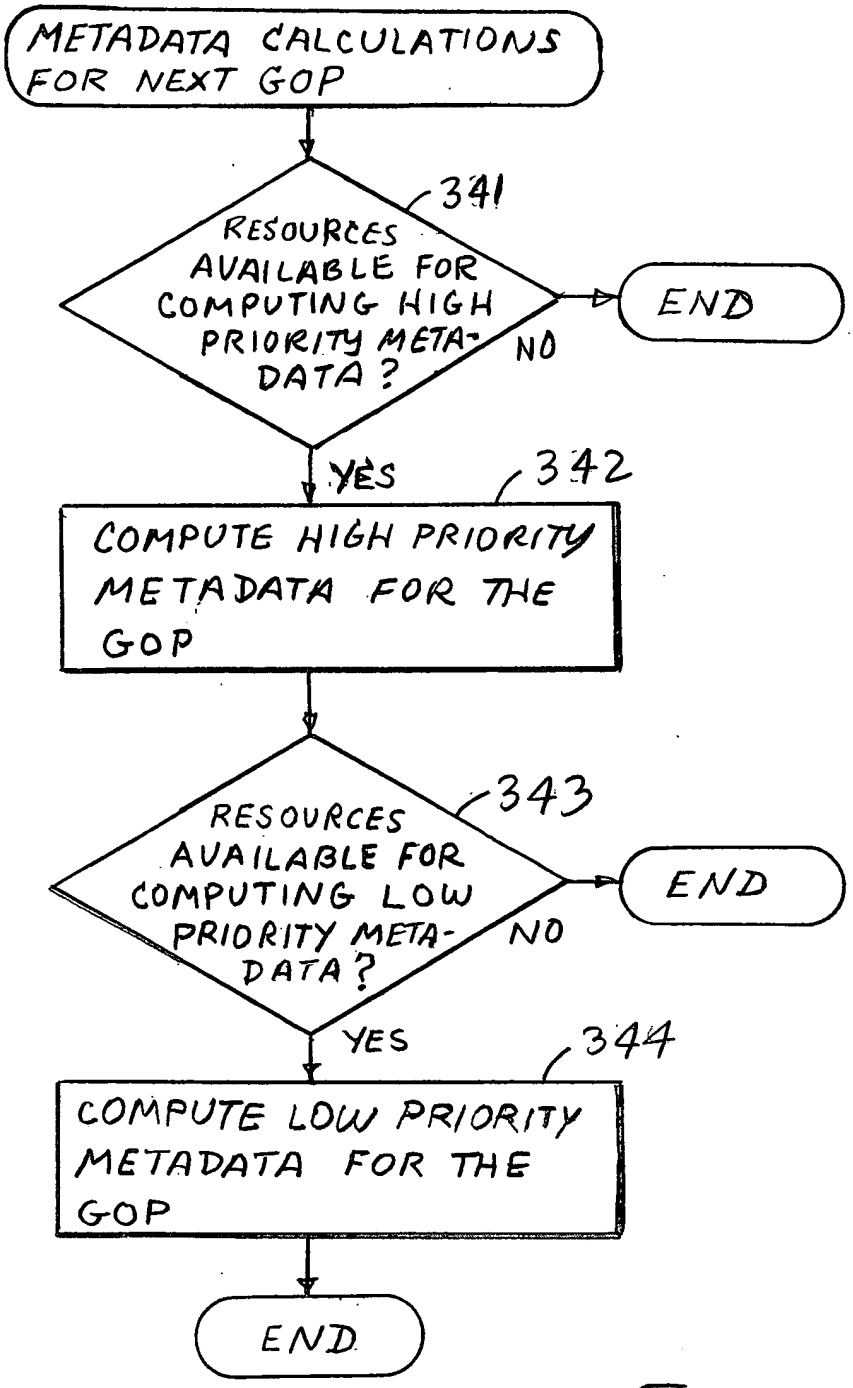
METADATA CALCULATIONS FOR NEXT GOP
341
RESOURCES AVAILABLE FOR COMPUTING HIGH PRIORITY META-DATA ?
NO
END
YES
342
COMPUTE HIGH PRIORITY METADATA FOR THE GOP
343
RESOURCES AVAILABLE FOR COMPUTING LOW PRIORITY META-DATA ?
NO
END
YES
344
COMPUTE LOW PRIORITY METADATA FOR THE GOP
END

FIG. 48

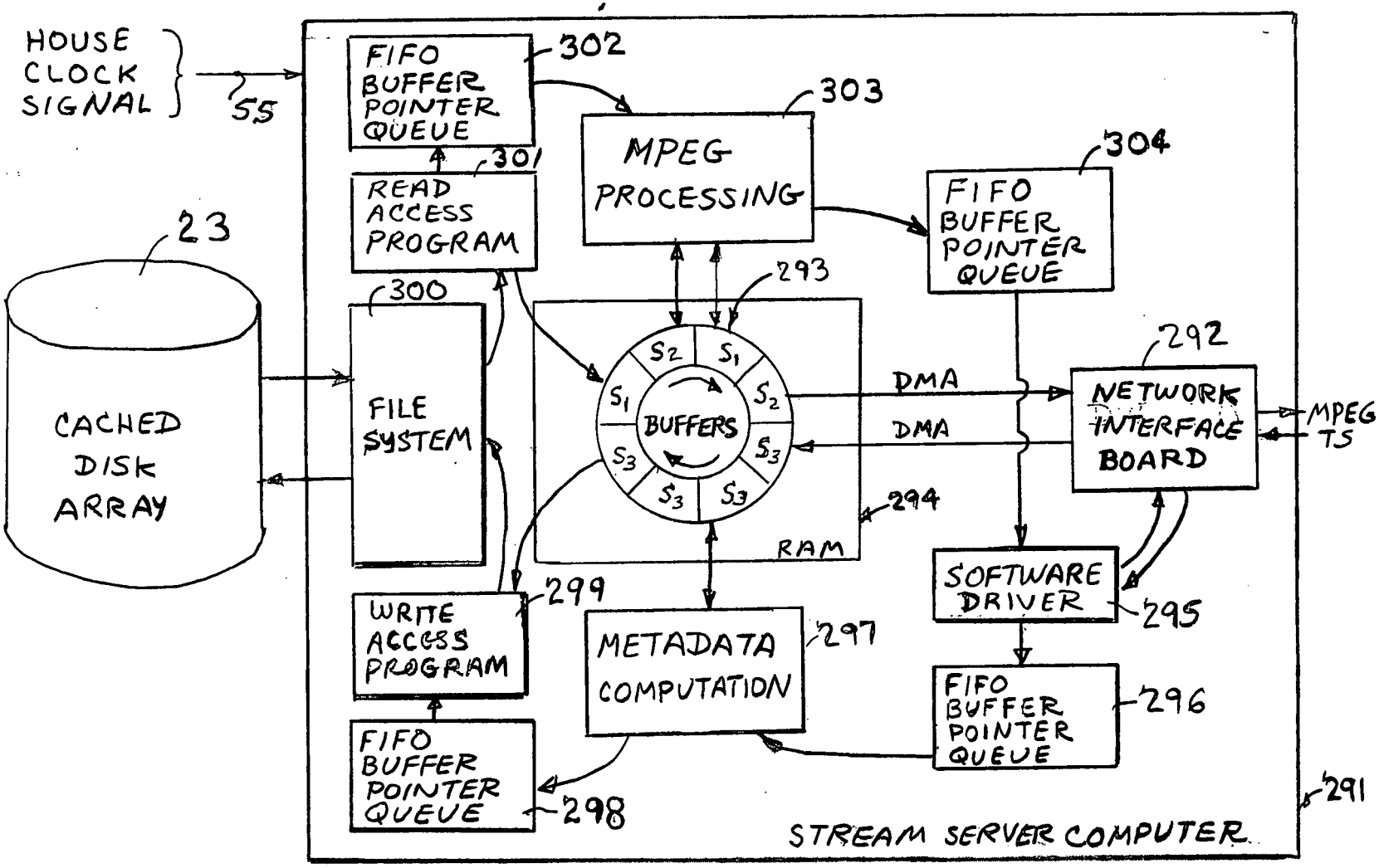
HOUSE CLOCK SIGNAL
55
FIFO BUFFER POINTER QUEUE
302
MPEG PROCESSING
303
FIFO BUFFER POINTER QUEUE
304
READ ACCESS PROGRAM
301
23
CACHED DISK ARRAY
300
FILE SYSTEM
293
S2
S1
S1
S2
BUFFERS
S3
S3
S3
S3
DMA
DMA
292
NETWORK INTERFACE BOARD
MPEG TS
294
RAM
SOFTWARE DRIVER
295
299
WRITE ACCESS PROGRAM
METADATA COMPUTATION
297
FIFO BUFFER POINTER QUEUE
296
FIFO BUFFER POINTER QUEUE
298
291
STREAM SERVER COMPUTER

FIG. 49

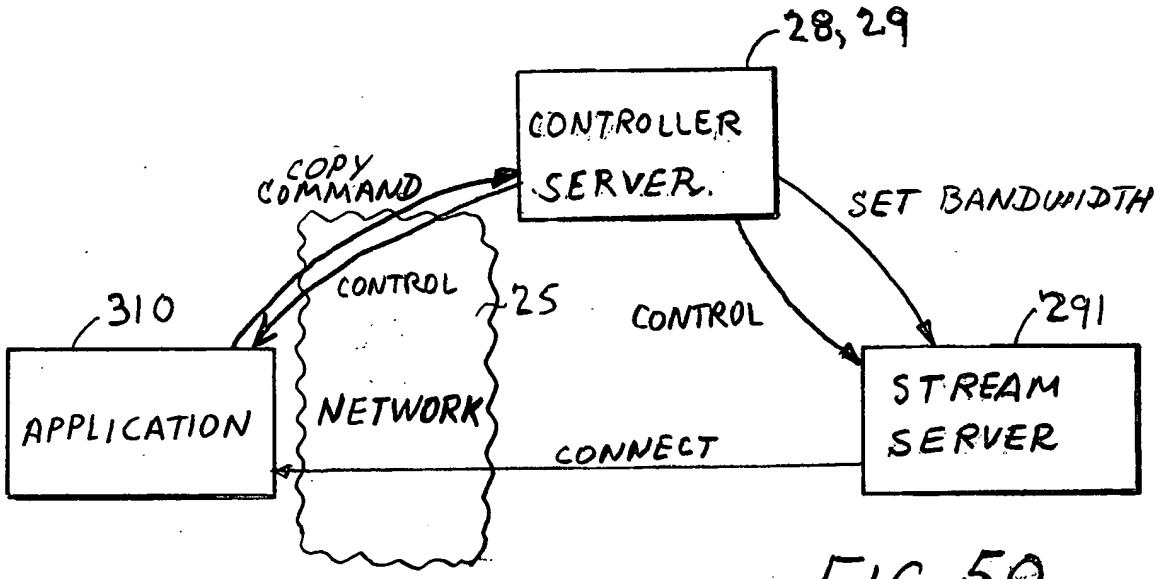
28, 29
CONTROLLER SERVER.
COPY COMMAND
SET BANDWIDTH
CONTROL
25
CONTROL
310
291
APPLICATION
NETWORK
STREAM SERVER
CONNECT

FIG. 50

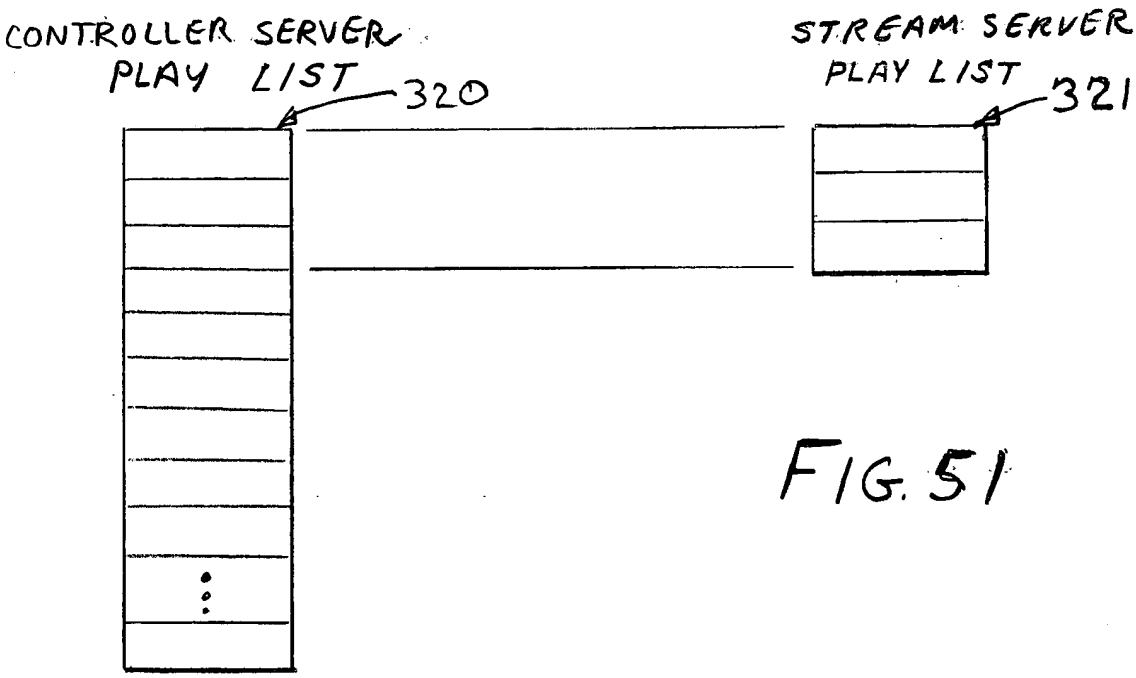
CONTROLLER SERVER PLAY LIST
320
STREAM SERVER PLAY LIST
321

FIG. 51

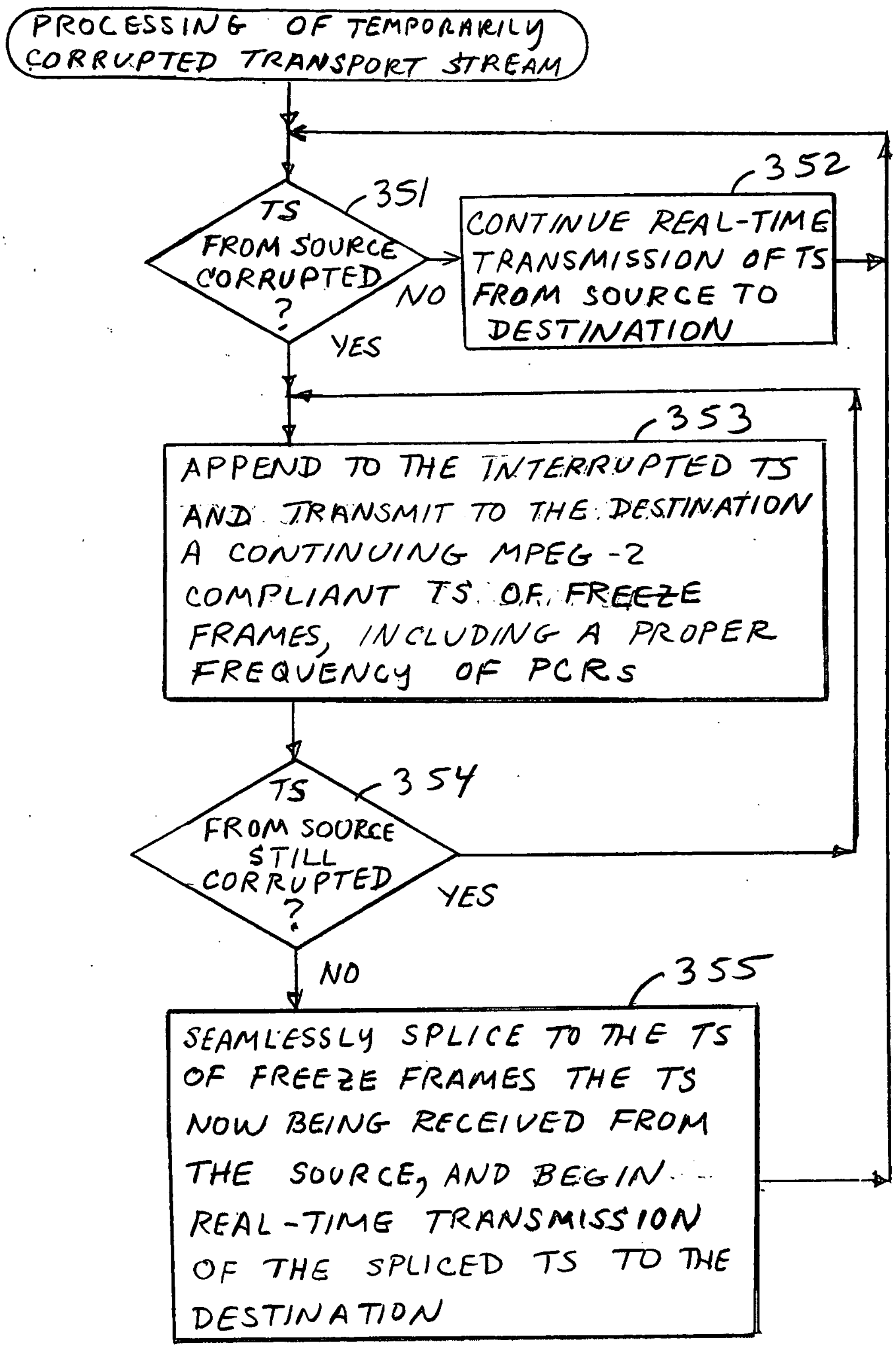
PROCESSING OF TEMPORARILY CORRUPTED TRANSPORT STREAM
TS FROM SOURCE CORRUPTED ?
351
NO
YES
CONTINUE REAL-TIME TRANSMISSION OF TS FROM SOURCE TO DESTINATION
352
APPEND TO THE INTERRUPTED TS AND TRANSMIT TO THE DESTINATION A CONTINUING MPEG-2 COMPLIANT TS OF FREEZE FRAMES, INCLUDING A PROPER FREQUENCY OF PCRs
353
TS FROM SOURCE STILL CORRUPTED ?
354
YES
NO
SEAMLESSLY SPLICE TO THE TS OF FREEZE FRAMES THE TS NOW BEING RECEIVED FROM THE SOURCE, AND BEGIN REAL-TIME TRANSMISSION OF THE SPLICED TS TO THE DESTINATION
355

FIG. 52